

APPLICATIONS OF BAYESIAN INFERENCE FOR THE ORIGIN  
DESTINATION MATRIX PROBLEM

by

Alara Güler

B.Sc., Manufacturing Systems, Sabancı University, 2015

M.Sc., Industrial Engineering, Sabancı University, 2018

Submitted to the Faculty of Engineering and Natural Sciences in partial fulfillment of  
the requirements for the degree of  
Master of Science

Graduate Program in Industrial Engineering  
Sabancı University  
2018

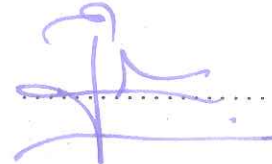
APPLICATIONS OF BAYESIAN INFERENCE FOR THE ORIGIN  
DESTINATION MATRIX PROBLEM

APPROVED BY:

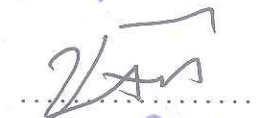
Assist. Prof. Dr. Sinan Yıldırım  
(Thesis Supervisor)



Prof. Dr. Ş. İlker Birbil  
(Thesis Co-supervisor)



Assist. Prof. Dr. Kamer Kaya



Assoc. Prof. Dr. Ali Taylan Cemgil



Prof. Dr. Berrin Yanıkoğlu



DATE OF APPROVAL: 10.01.2018



## ACKNOWLEDGEMENTS

I would like to thank my supervisors Dr. Sinan Yıldırım and Prof. Dr. İlker Birbil for their help, guidance, and patience along with their knowledge. I would like to also thank my friends and parents for their support. This thesis is dedicated to my parents Dr. Canan Güler and Dr. Aykut Güler.

## ABSTRACT

### APPLICATIONS OF BAYESIAN INFERENCE FOR THE ORIGIN DESTINATION MATRIX PROBLEM

This thesis presents a study of estimating the probability matrix of an origin-destination model associated with a two-way transportation line with the help of Bayesian inference and Markov chain Monte Carlo methods, more specifically, Metropolis within Gibbs algorithm. Collecting the exact count data of a transportation system is often not possible due to technical insufficiencies or data privacy issues. This thesis concentrates on the utilization of Markov chain Monte Carlo Methods for two origin-destination problems: one that assumes missing departure data and one that assumes the availability of differentially private data instead of the complete data. Different models are formulated for those two data conditions that are under study. The experiments are conducted with synthetically generated data and the performance of each model under these conditions were measured. It has been concluded that MCMC methods can be useful for effectively estimating the probability matrix of certain OD problems.

## ÖZET

### KÖKENLİ VARİŞ NOKTASI PROBLEMLERİNE YÖNELİK BAYESÇİ ÇIKARIM UYGULAMALARI

Bu tez, çift yönlü ve tek hatlı bir metro sistemiyle ilişkili kökenli varış probleminin olasılık matrisini Bayesci çıkarım ve Markov zinciri Monte Carlo metodları kullanarak kestiren bir çalışma sunmaktadır. Bir ulaşım sisteminin kesin sayım verilerini toplamak çoğu zaman teknik eksiklikler ve veri gizliliği politikaları sebebiyle mümkün olmamaktadır. Bu tezin odağı eksik veri toplandığı veya gürültülü veri yayımlandığı koşullarda, İstanbul'daki Kadıköy-Pendik metro hattına benzer, iki yönlü tek hatlı metro sistemlerinin olasılık matrisini Markov zinciri Monte Carlo metodlarını kullanarak kestirmektir. Eksik ve gürültülü veri elde edildiği durumlarda kullanılabilecek değişik modeller formüle edilmiştir. Veri sağlayıcıdan gerçek veri elde edilemediği için veri sentetik olarak tarafımızca oluşturulmuş ve formüle edilen modellerin olasılık matrisini kesirmekteki performansları değerlendirilmiştir. Markov zinciri Monte Carlo metodlarının konumuz olan kökenli varış problemlerinin olasılık matrisini etkin bir şekilde kestirmekte kullanılabileceği sonucuna ulaşılmıştır.

# TABLE OF CONTENTS

ACKNOWLEDGEMENTS . . . . .	iii
ABSTRACT . . . . .	iv
ÖZET . . . . .	v
LIST OF FIGURES . . . . .	viii
LIST OF TABLES . . . . .	x
LIST OF SYMBOLS . . . . .	xi
LIST OF ACRONYMS/ABBREVIATIONS . . . . .	xiii
1. INTRODUCTION . . . . .	1
1.1. The Origin-Destination (OD) Matrix Problem . . . . .	1
1.1.1. Incomplete Data Collection . . . . .	2
1.1.2. Data Privacy Issues . . . . .	4
1.2. Differential Privacy . . . . .	4
1.2.1. Laplace Mechanism . . . . .	6
1.3. Markov Chain Monte Carlo (MCMC) and Metropolis-Hastings within Gibbs Algorithm . . . . .	8
2. BAYESIAN ODM PARAMETER ESTIMATION USING ENTRY-ONLY DATA	11
2.1. The Entry Only Data and the Model . . . . .	11
2.2. The Inferential Problem . . . . .	13
2.3. Methodology . . . . .	13
2.3.1. Markov chain Monte Carlo . . . . .	16
2.3.1.1. Sampling $D_{1:M}$ . . . . .	16
2.3.1.2. Sampling $\rho$ matrix . . . . .	17
2.3.1.3. Updating $\alpha$ . . . . .	17
2.4. Results . . . . .	19
2.5. Discussion . . . . .	23
3. BAYESIAN ODM PARAMETER ESTIMATION USING THE NOISY DATA	25
3.1. The Available Data and the Model . . . . .	25
3.2. The Inferential Problem . . . . .	28
3.3. Methodology . . . . .	28

3.3.1.	Markov Chain Monte Carlo - Type 1 . . . . .	31
3.3.1.1.	Updating $(x_{1,i}, \dots, x_{D,i})$ and $(h_{1,i,:}, \dots, h_{D,i,:})$ . . . . .	31
3.3.1.2.	Sampling $\lambda_i$ . . . . .	33
3.3.1.3.	Sampling $\rho_{i,:}$ . . . . .	33
3.3.2.	Markov Chain Monte Carlo - Type 2 . . . . .	34
3.3.3.	Markov Chain Monte Carlo - Type 3 . . . . .	35
3.3.4.	Markov Chain Monte Carlo - Type 4 . . . . .	37
3.4.	Discussion and Results . . . . .	39
4.	CONCLUSION . . . . .	51
	REFERENCES . . . . .	54

## LIST OF FIGURES

2.1	Comparison of the estimated $\alpha$ with its true value through iterations when $\alpha = 0.001$ . . . . .	20
2.2	Comparison of the estimated $\rho$ with their true values for $\alpha = 0.001$	20
2.3	Comparison of the estimated $\alpha$ with its true value through iterations when $\alpha = 0.0001$ . . . . .	21
2.4	Comparison of the estimated $\rho$ with their true values for $\alpha = 0.0001$	21
2.5	Comparison of the estimated $\alpha$ with its true value through iterations when $\alpha = 0.01$ . . . . .	22
2.6	Comparison of the estimated $\rho$ with their true values for $\alpha = 0.01$	22
2.7	Comparison of the MSE values for different $\log \alpha$ levels . . . . .	24
3.1	Comparison of the $\lambda$ samples generated by model Type 1 with their true value through iterations when $\epsilon = 1$ . . . . .	43
3.2	Comparison of the $\rho$ samples generated by model Type 1 with their true values for $\epsilon = 1$ . . . . .	43
3.3	Comparison of the $\lambda$ samples generated by model Type 1 with their true value through iterations when $\epsilon = 2$ . . . . .	44
3.4	Comparison of the $\rho$ samples generated by model Type 1 with their true values for $\epsilon = 2$ . . . . .	44
3.5	Comparison of the $\lambda$ samples generated by model Type 1 with their true value through iterations when $\epsilon = 5$ . . . . .	45
3.6	Comparison of the $\rho$ samples generated by model Type 1 with their true values for $\epsilon = 5$ . . . . .	45
3.7	Comparison of the $\lambda$ samples generated by model Type 1 with their true value through iterations when $\epsilon = 10$ . . . . .	46
3.8	Comparison of the $\rho$ samples generated by model Type 1 with their true values for $\epsilon = 10$ . . . . .	46
3.9	Comparison of the $\lambda$ samples generated by model Type 2 with their true value through iterations when $\epsilon = 1$ . . . . .	47

3.10	Comparison of the $\rho$ samples generated by model Type 2 with their true values for $\epsilon = 1$ . . . . .	47
3.11	Comparison of the $\lambda$ samples generated by model Type 2 with their true value through iterations when $\epsilon = 2$ . . . . .	48
3.12	Comparison of the $\rho$ samples generated by model Type 2 with their true values for $\epsilon = 2$ . . . . .	48
3.13	Comparison of the $\lambda$ samples generated by model Type 2 with their true value through iterations when $\epsilon = 5$ . . . . .	49
3.14	Comparison of the $\rho$ samples generated by model Type 2 with their true values for $\epsilon = 5$ . . . . .	49
3.15	Comparison of the $\lambda$ samples generated by model Type 2 with their true value through iterations when $\epsilon = 10$ . . . . .	50
3.16	Comparison of the $\rho$ samples generated by model Type 2 with their true values for $\epsilon = 10$ . . . . .	50

## LIST OF TABLES

2.1	Example Data Regarding One Card . . . . .	11
2.2	$MSE_1$ and $MSE_2$ values for different values of $\alpha$ . . . . .	23
3.1	An example $H_u$ where $n = 3$ . . . . .	25
3.2	Values of IAC times yielded by each model under different $\epsilon$ values	40
3.3	Values of IAC times yielded by models in Scenario 2 under different $\epsilon$ values . . . . .	41
3.4	Mean Value of Norm of the Difference Matrix of $\rho$ for Type 1 and Type 2 . . . . .	42
3.5	Mean Value of Norm of the Difference Matrix of $\lambda$ for Type 1 and Type 2 . . . . .	42



## LIST OF SYMBOLS

$A$	The station that the passenger arrives
$B$	The station that the passenger arrives next
$D$	The station that the passenger exits the system <i>or</i> Total number of noisy $H$ matrixes provided by the data holder
$\exp$	Power of the natural exponential constant $e$
$g_\alpha(\cdot \cdot, \cdot)$	The probability of arriving at a spesific station given $D$ , $T_D$ , and $T_B$
$h_{d,i,j}$	$(i, j)$ 'th element of the $H$ matrix related to day $d$
$H$	The origin-destination matrix
$\tilde{h}_{d,i,j}$	$(i, j)$ 'th element of the $\tilde{H}$ matrix related to day $d$ in the missing data scenario
$\tilde{H}$	The noisy origin-destination matrix provided by the data holder in the noisy data scenario
$k$	Index of the iterations in algorithms
$M$	The total number of entries in the missing data scenario
$n$	Number of stations in a metro-line
$p(\cdot \cdot)$	Probability density
$q(\cdot \cdot)$	Conditional density of the proposal kernel
$Q(\cdot \cdot)$	Proposal kernel
$T_A$	The time of arrival to station $A$
$T_B$	The time of arrival to station $B$
$y$	The data regarding one user
$Y$	The total data set
$x_{d,i}$	Total number of passengers that arrived to station $i$ in day $d$
$\alpha$	Scale parameter of the Gamma distribution <i>or</i> self return parameter (as indicated in thesis)
$\beta$	Shape parameter of the Gamma distribution
$\delta$	Parameters of a Dirichlet distribution

$\epsilon$	Privacy factor
$\lambda$	Rate parameter of the Poisson distribution
$\mu$	Mean of a probability distribution
$\eta$	Logarithm of the self-return parameter $\alpha$
$\pi_d(\cdot \cdot)$	Full conditional distribution of the $d$ 'th component
$\rho_{i,j}$	$(i, j)$ 'th element of the probability matrix
$\rho$	The probability matrix of the OD problem
$\sigma$	Standard deviation of a probability distribution
$\theta$	Set of unknown parameters

## LIST OF ACRONYMS/ABBREVIATIONS

IAC	Integrated Auto Correlation
MCMC	Markov chain Monte Carlo
MH	Metropolis-Hastings
MSE	Mean Squared Error
OD	Origin-Destination
ODM	Origin-Destination Matrix

# 1. INTRODUCTION

## 1.1. The Origin-Destination (OD) Matrix Problem

The origin-destination matrix (ODM) problem involves finding passenger preferences in a transportation line. Let  $n > 1$  be the number of stations in a transportation line and  $H$  be a  $n \times n$  ODM associated with this transportation line. The  $(i, j)$ 'th element of the origin destination matrix  $H$ ,  $h_{i,j}$ , denotes the number of customers that enter the metro line from the  $i$ 'th station and depart from the  $j$ 'th station. In this thesis, it is assumed that for each customer that enters the metro line at station  $i$ , the probability of leaving the line at station  $j$  is denoted by  $\rho_{i,j}$ . Let  $\rho$  be the matrix of these probabilities with the  $(i, j)$ 'th element being  $\rho_{i,j}$ .

The rows of the  $H$  can be normalized to yield a maximum likelihood estimation of the probability matrix  $\rho$ . Note that, since one cannot enter and depart from the same station, the diagonals of both  $H$  and  $\rho$  are zero. Estimating the  $\rho$  matrix directly would be straightforward if the  $H$  matrix was fully observable, however most of the time, these matrices can not be obtained directly due to various mechanisms which result in incomplete or noisy data. Collecting the whole data would be possible by recording each passenger's entrance and exit stations. If each passenger's entrance and exit stations were recorded, it would mean that each passenger's travel information is also recorded. Knowing the routes and journeys of all passengers would lead to computing the  $\rho$  matrix directly. Since collecting the complete data would require collecting each individuals data separately, a technical infrastructure which collects this data should be present, therefore each passenger needs to be registered to this system. Especially, when it comes to estimating the  $\rho$  matrix of the traffic data, this infrastructure is not present, however estimation to an extent is still possible. In some cases, the original data might still not be available even if they were collected completely. This situation might be faced if the data holders' policy is to release a differentially private data in order to protect the privacy of the passengers. This policy causes the data obtained by the data holder to be noisy, therefore direct analysis from

the obtained data will not yield the most realistic results. This thesis focuses on and suggests methods for estimating the  $\rho$  matrix of a two-way metro-line which is similar to the Istanbul railway system under two different cases in which the original  $H$  matrix can not be obtained due to:

- incomplete data collection
- data privacy issues

which then leads to missing and noisy data conditions respectively.

### 1.1.1. Incomplete Data Collection

Previous studies conducted under incomplete data environment involved utilizing passenger surveys (Watling, 1994), and traffic counts, where the statistical approaches such as maximum likelihood were discussed (Cascetta et al., 1993; Cascetta and Nguyen, 1988). However Bayesian statistics, which gives weight to prior beliefs and available data, were explored deeply since as early as 1983 (Maher, 1983) with available traffic link counts information. In a later work by Tebaldi and West (1998), Bayesian statistics were proposed to be a feasible approach to such origin-destination problems with missing data including the traffic flow rates, link counts, and prior outdated estimates of the matrices. Bayesian inference framework was investigated further by Li with the addition of the Expectation Maximization algorithm which reduced the computational effort required to compute the posterior (Li, 2005). Hazelton (2001) suggested that estimating and predicting O-D matrices with the help of Markov chain Monte Carlo methods, more specifically Metropolis-Hastings algorithm, have great potential compared to reconstructing methods. Ni and Leonard II (2005) later proposed using Markov chain Monte Carlo methods in order to impute, simulate, and sample the missing data and analyze the estimation problem with the help of resampled data. Their work is important because they have showed that Markov chain Monte Carlo methods were successful and accurate to estimate the traffic count, speed, and, density of the system when the complete data was not available. We will utilize a similar approach in order to deal with our missing data model; however, our aim is not to

simulate and impute the missing data only, but also to estimate the  $\rho$  matrix, the mechanism that lies behind the data.

Estimating the  $\rho$  matrix of some railway systems is easier since some governments adopted the Smart Card system which either collects the complete data, such as Netherlands Railway System, or the partial data, such as Istanbul Railway System. In Istanbul, a smart-card issued by the government called Istanbul Kart is widely used by locals. It is possible to purchase and load credits and use these credits when entering public transportation. When a passenger enters the metro line, he scans his card at a machine which then reduces the credits in the card. However, they do not re-scan their cards when they leave the metro-line. Thus, the data collected on the cards contains only the entrance information to the metro line, not the exit information. Since the data regarding the exit information is not available, it is not possible to directly obtain  $H$ , and therefore a direct estimate for the matrix  $\rho$  from  $H$ . A counting method which accepts the departure station to be the second of two consecutive arrivals based on the assumption that a daily passenger will not use other means of transportation between two consecutive entries was proposed by Zhao and Rahbee (2007). However the assumptions made by Zhao and Rahbee (2007) model the exit stations of the customers deterministically rather than stochastically, hence statistical inference methods were not utilized. Another similar work which estimates the destination points in a missing data environment was conducted by Munizaga and Palma (2012) where the destination point on a metro line is guessed by the combined information gathered from the Smart Cards and the GPS data of the Santiago, Chile transportation system. The missing data points were filled through Markov chain Monte Carlo methods and statistical analysis on the completed data were conducted. Later, this work was validated to estimate the OD matrix up to 90 percent correctness (Munizaga et al., 2014). Since our missing data model assumes that only the Smart Card data with missing exit station information are available, we decided to fit a probabilistic model for the exit stations and simulated these missing data points with the help of Markov chain Monte Carlo methods.

### 1.1.2. Data Privacy Issues

As mentioned previously, collecting the complete data means collecting and storing each individual's data. In such cases where the whole data was collected, the original  $H$  matrix might still not be available to the researchers due to differential privacy issues. The related institution is likely to provide an altered version of the  $H$  matrix, which adds noise to the original  $H$  matrix, so that the privacy of each passenger is protected since it would not be ethical to issue each person's travel data.

There are rather small number of work related to parameter estimation and statistical inference for the differentially private data. The first work that addresses the parameter estimation problem from a differentially private data was conducted by Charest (2010). In this work it was concluded that it is possible to infer the parameters of a differentially private data through Markov chain Monte Carlo methods if the parameters of the privacy mechanism is taken into consideration. Charest (2010) also provided an example experiment in which they conducted analysis on the posterior distributions of the Beta-Binomial mechanism. Although there is yet no work present on the estimation of ODM parameters from differentially private data, Markov chain Monte Carlo methods were proven to be a viable approach for estimation under differential privacy (Lu and Miklau, 2014). In this thesis implementation of the Markov chain Monte Carlo methods for differentially private ODM will be explored.

## 1.2. Differential Privacy

Differential Privacy concept has emerged to address the concern of releasing data regarding the individuals in large data sets. As mentioned previously, collecting the necessary data in order to conduct statistical analysis requires collecting each individual's data. Most of the private data of an individual, such as health record, travel information, purchases etc. are collected through electronic systems and databases implemented by service providers. The collection of these data is mostly subject to further statistical analysis in order to provide insights about the population. However, releasing this data directly to third parties for analysis is either not possible due to

ethical and legal contracts or will give away important private information in individual level, therefore the institutions that possess this data are motivated to protect the privacy of each individual when it comes to publishing this data. Differential Privacy is a set of methods and algorithms devoted to protect each individual's data while allowing statistical analysis from the data as a whole (Dwork and Roth, 2014). Differential Privacy algorithms aim to alter the collection of the data in such a way that reaching to an individual's data is not possible while the whole data's implication is still viable. This mechanism is achieved by adding randomness to the collection of the data so that reaching to a definite conclusion about any individual is not possible.

There are a few definitions to be made in order to fully understand how Differential Privacy algorithms work. These definitions have been made by Dwork and Roth (2014) where they thoroughly formulated the fundamentals, algorithms, and applications of Differential Privacy. We need the following three definitions prior to formulating a differential privacy of an algorithm.

- Randomized algorithm
- Probability Simplex
- Distance Between Databases

**Definition 1** (Randomized Algorithm). *A randomized algorithm is a mechanism,  $M$ , that produces a mapping,  $M : A \rightarrow \Delta(B)$ , where  $A$  denotes the domain of the algorithm,  $B$  denotes a discrete range, and  $\Delta(B)$  denotes the probability simplex over  $B$ .*

**Definition 2** (Probability simplex). *The probability simplex,  $\Delta(B)$  is defined as follows:*

$$\Delta(B) = \left\{ \mathbf{x} \in \mathbb{R}^{|B|} : \mathbf{x}_i \geq 0, \forall i \text{ and } \sum_{i=1}^{|B|} \mathbf{x}_i = 1 \right\}.$$

In this notation,  $\mathbf{X}$  represents the data universe where each  $\mathbf{x}$  is collected from, where  $\mathbf{x}$  denotes the histogram of the data. Namely,  $\mathbf{x}_i$  denotes the number of type



$i$  elements present in the universe  $\mathbf{X}$  in the database  $\mathbf{x}$ . As the logic suggests,  $\mathbf{x}$  can take values from the non-negative integers set.

**Definition 3** (Hamming distance). *The Hamming distance between a couple of databases  $\mathbf{x}$  and  $\mathbf{y}$  is the number of entries that are different than each other. Hamming distance can be calculated for data sets which are equal in length. In other words,  $\text{Ham}(\mathbf{x}, \mathbf{y})$  can be found by comparing each corresponding entry with each other and recording the number of total different entities.*

Finally Dwork and Roth (2014) define Differential Privacy as:

**Definition 4** (Differential privacy). *A randomized algorithm,  $M$ , with its domain being  $\mathbb{N}^{|\mathbf{X}|}$  and where  $\mathbb{N}$  represents the set of non-negative integers, is  $(\epsilon, \delta)$  differentially private if  $\forall S \subseteq \text{Range}(M)$  and  $\forall \mathbf{x}, \mathbf{y} \in \mathbb{N}^{|\mathbf{X}|}$  such that  $\text{Ham}(\mathbf{x}, \mathbf{y}) \leq 1$ :*

$$\mathbb{P}(M(\mathbf{x}) \in S) \leq e^\epsilon \mathbb{P}(M(\mathbf{y}) \in S) + \delta$$

The quantity  $\epsilon$  is referred as the privacy factor that the algorithm provides and is a positive real number.  $\epsilon$ -differential privacy is a special case of differential privacy where  $\delta = 0$ .  $\epsilon$ -differential privacy ensures that:

$$\frac{\mathbb{P}(M(\mathbf{x}) = m)}{\mathbb{P}(M(\mathbf{y}) = m)} \leq e^\epsilon.$$

### 1.2.1. Laplace Mechanism

The data we investigate in this thesis is an example of counting data since the elements of the  $H$  matrix,  $H_{i,j}$ 's, denote the number of passengers which entered the railway system from station  $i$  and departed from station  $j$ . A mechanism called the Laplace mechanism is widely used to ensure the differential privacy of each individual's data in a counting data set. Laplace mechanism adds a Laplacian noise to each entry in a counting data and issues a noisy version of the original count data. The Laplace

mechanism applied to a function  $f : X \rightarrow \mathbb{R}$  of the data is defined by (Dwork and Roth, 2014) as follows:

$$M_L(x, f(\cdot), \epsilon) = f(x) + (Y_1, \dots, Y_k).$$

Here,  $Y_i$ s are random variables drawn from  $\text{Lap}(\frac{\Delta f}{\epsilon})$  where  $\Delta f$  represents the sensitivity. The sensitivity is defined as follows:

$$\Delta f = \max_{y, y' : \|y - y'\| \leq 1} |f(y) - f(y')|.$$

The sensitivity of the data helps to determine the noise that should be added in order to ensure every user's privacy. Sensitivity can be perceived as the maximum possible contribution of an individual to the whole data, therefore adding noise according to this measure will prevent capturing data in individual level. The Laplace Mechanism can be used to produce an  $\epsilon$ -differentially private data from counting and histogram databases (Dwork and Roth, 2014). There are other methods such as the Exponential Mechanism and the Gaussian Mechanism in order to generate differentially private data, however these methods are out of the scope this thesis.

In Chen et al. (2014), differential privacy via the Laplace mechanism is ensured in order to avoid jeopardising passengers individual data in transportation systems. In this thesis, the noisy data model is assumed to be protected with the Laplace mechanism and feasibility of the Metropolis-Hastings within Gibbs algorithm in order to estimate the  $\rho$  matrix is explored. The literature review revealed that, a work which employs Markov chain Monte Carlo estimation methods for transportation systems and OD matrices in a differentially private environment is not yet present.

### 1.3. Markov Chain Monte Carlo (MCMC) and Metropolis-Hastings within Gibbs Algorithm

An ergodic Markov chain whose stationary distribution is  $\pi$  will converge to  $\pi$  if it is simulated for a relatively long time. If it is possible to design such a Markov chain which has the stationary distribution as the desired distribution of the estimated parameters, then it is also possible to run this Markov chain for a long enough time and sample the estimated parameters from this distribution (Yıldırım, 2016). This method has been proven to converge in the previous literature. The foundations of the Markov chain Monte Carlo methods were first built by Metropolis and Ulam (1949) and then improved by Hastings (1970).

One of the most popular Markov chain Monte Carlo algorithms is the Metropolis-Hastings (MH) algorithm and this algorithm has been studied quite widely in the literature in different application areas. The Metropolis-Hastings algorithm is an iterative process which proposes a new value for the estimated parameter depending on its previous value in each iteration. The proposed value is accepted with a certain probability called the acceptance probability  $\alpha(X^{(k-1)}, X')$  and if accepted, the current value of the parameter is updated to the proposed value  $X'$  (Hastings, 1970). The algorithm starts by initializing  $X^{(1)}$  from some beginning point, then the steps conducted at iteration  $k, k > 1$  is given in Algorithm 1.

---

**Algorithm 1:** Metropolis-Hastings Algorithm at  $k$ 'th iteration

---

- 1 Sample  $X' \sim Q(\cdot|X^{(k-1)})$
  - 2 Set  $X^{(k)} = X'$  with probability  $\alpha(X^{(k-1)}, X')$ ; otherwise set  $X^{(k)} = X^{(k-1)}$
- 

Let  $q(\cdot|\cdot)$  be the conditional density of the proposal kernel  $Q(\cdot|\cdot)$ . The acceptance probability,  $\alpha(X^{(k-1)}, X')$ , is defined as follows:

$$\alpha(X^{(k-1)}, X') = \min \left[ 1, \frac{\pi(X')q(X^{(k-1)}|X')}{\pi(X^{(k-1)})q(X'|X^{(k-1)})} \right].$$

There are two important special cases of the design of the  $Q$ . The first of these is the

symmetric choice for  $Q$  so that  $q(X'|X^{(k-1)}) = q(X^{(k-1)}|X')$ . This special case is called the random walk MH. Since  $q(X'|X^{(k-1)}) = q(X^{(k-1)}|X')$ , the acceptance probability for this model becomes:

$$\alpha(X^{(k-1)}, X') = \min \left[ 1, \frac{\pi(X')}{\pi(X^{(k-1)})} \right].$$

The second of the special cases is called the independence Metropolis-Hastings algorithm. For this algorithm,  $Q$  is designed in such a way that the proposed value  $X'$  does not depend on its previous value, namely  $q(X'|X^{(k-1)}) = q(X')$ . In this case, it is easy to see that the acceptance probability becomes:

$$\alpha(X^{(k-1)}, X') = \min \left[ 1, \frac{\pi(X')q(X^{(k-1)})}{\pi(X^{(k-1)})q(X')} \right].$$

Selection of  $Q$  determines the efficiency of the algorithm, therefore designing an appropriate  $Q$  is of core importance for the algorithm. Metropolis-Hastings algorithm is quite useful for targeting posterior distributions of parameters whose joint distributions are normally intractable but can be computed up to a proportionality constant. Therefore selection of  $Q$  plays a very important role in this thesis and its applications will further be examined in the coming chapters where the models are discussed in more detail.

Another widely used MCMC algorithm is the Gibbs sampler (Gelfand and Smith, 1990; Geman and Geman, 1984). This algorithm is particularly useful when  $X$ , the estimated parameters, is multi-dimensional, has  $D > 1$  components such as  $X = (X_1, \dots, X_D)$ , as it allows sampling each component by their conditional densities, namely full conditional distributions  $\pi_d$ , depending on the other components (Yildirim, 2016). Gibbs sampling is also an iterative process: At each iteration, each component is sampled and updated depending on the full conditional distributions on the other parameters. Since components of  $X$  are sampled from full conditional distributions, there is no concept of acceptance probability as there is in the Metropolis-Hastings algorithm (Gelfand and Smith, 1990; Geman and Geman, 1984). Similar to Metropolis-

Hastings algorithm, Gibbs algorithm starts by initializing  $X^{(1)}$  at some beginning point. The steps conducted at iteration  $k, k > 1$  is given in Algorithm 2.

---

**Algorithm 2:** Gibbs Sampling Algorithm at the  $k$ 'th iteration

---

1 Sample  $X_d^k \sim \pi_d(\cdot | X_{1:d-1}^k, X_{d+1:D}^{k-1})$  for  $d = 1, \dots, D$

---

It can be clearly seen from the algorithm that, sampling  $X_d^k$  requires all conditional distributions of the components to be computed. In some cases, the full conditional density of the  $d$ 'th component,  $\pi_d(\cdot | X_{1:d-1}^k, X_{d+1:D}^{k-1})$ , is not tractable. In such cases, it is still valid to replace the Gibbs sampler move for the  $d$ 'th step with a Metropolis-Hastings move that updates this component,  $X_d^k$  that targets  $\pi_d(\cdot | X_d^{k-1})$  (Yıldırım, 2016). This alteration in the algorithm is called Metropolis-Hastings within Gibbs algorithm and the generalized form is given in Algorithm 3.

---

**Algorithm 3:** Metropolis-Hastings within Gibbs algorithm at the  $k$ 'th iteration

---

1 Update  $X_d^{(k)}$  by using a Metropolis-Hastings move that targets  $\pi_d(\cdot | X_{1:d-1}^k, X_{d+1:D}^{k-1})$  for  $d = 1, \dots, D$

---

We have utilized this property of these two algorithms in all of our models for the parameters whose full conditionals were not tractable. In Chapters 2 and 3, the detailed formulation of the model used to estimate the unknown parameters for the Entry Only Data scenario and the results yielded by this model are given, in Chapter 3, the detailed formulation of the model and comparison of 4 different proposal densities used to estimate the unknown parameters for the Noisy Data scenario and results yielded by this model are given. The thesis is then concluded in Chapter 4 and possible improvements for the future work are discussed.

## 2. BAYESIAN ODM PARAMETER ESTIMATION USING ENTRY-ONLY DATA

### 2.1. The Entry Only Data and the Model

This model simulates a single line two-way transportation system. As an example, we will consider the metro line in the Anatolian side of Istanbul. As stated above, passengers of Istanbul metro-line use a smart-card called Istanbul Kart in order to enter the metro system. This system only collects data when entering the metro-line but does not collect information about the destination of the passenger, the station from which the passenger leaves the system. We assume a single line with  $n > 1$  stations, where passengers can travel in both directions. The available data in our problem contain only the arrival information for passengers since the cards do not hold any data regarding the exit information. A portion of an example of the available data can be seen in Table 2.1.

Table 2.1. Example Data Regarding One Card

Card ID	Time of Arrival	Arrival Station
1	07.42	5
1	11.48	9
1	12.10	9
1	18.03	2
$\vdots$	$\vdots$	$\vdots$

From such a data set, we can deduce the following variables for a trip that a customer performs:

- Passenger ID,
- The stop the passenger arrives,  $A \in \{1, \dots, n\}$ ,
- The time of arrival,  $T_A$ ,
- The stop the passenger arrives next,  $B \in \{1, \dots, n\}$ ,
- The time of the next arrival,  $T_B$ .

In an entry-only data set, the missing information is the departure time and the station that the passenger leaves the system from. Each passenger arrives to the system at a random station  $A$  at a random time. We will denote the departure station by the random variable  $D$  and the time of departure by  $T_D$ . In the origin-destination problem, the conditional probability distribution of  $D$  given  $A$  is represented by an  $n \times n$   $\rho$  matrix, where

$$\mathbb{P}(D = d|A = a) = \rho_{a,d}.$$

For simplicity, it is assumed that the travel time between each station is the same and it is  $\Delta$ ; therefore, the departure time is calculated as follows:

$$T_D = T_A + |D - A|\Delta. \quad (2.1)$$

Our model assumes that, as the time between the departure and the next arrival increases, the probability of re-entering the system from the departure station decreases. The reasoning behind this assumption is that it is highly likely that if a passenger spends a large amount of time after departing, the probability of using other means of transportation increases, and therefore they are less likely to come back to the same station from which they departed. This assumption is reflected by our probability model for the next station of arrival: The probability of arriving at station  $B = b$  given  $D$ ,  $T_D$  and  $T_B$  is denoted as:

$$\mathbb{P}(B = b|D = d, T_D = t_D, T_B = t_B) = \begin{cases} g_\alpha(b|d, t_B - t_D), & t_B > t_D \\ 0, & t_B < t_D. \end{cases} \quad (2.2)$$

If we let  $\tau$  denote  $t_B - t_D$ , for  $\tau > 0$ , we can calculate  $g_\alpha(b|d, \tau)$  as

$$g_\alpha(b|d, \tau) = \begin{cases} \frac{\exp\{-\alpha/\tau\}}{1+(n-1)\exp\{-\alpha/\tau\}}, & b \neq d \\ \frac{1}{1+(n-1)\exp\{-\alpha/\tau\}}, & b = d. \end{cases} \quad (2.3)$$

This constructs the conditional probability of the next station of arrival  $B$  given the first station of arrival  $A$ ,

$$\mathbb{P}(B = b|A = a, T_B = t_B, T_A = t_A) = \sum_{d=1}^n \rho_{\alpha,d} g_{\alpha}(b|d, \tau) \quad (2.4)$$

where  $t_D$  is calculated from  $t_A$ ,  $\alpha$  and  $d$  using the relation in (2.1).

## 2.2. The Inferential Problem

Without loss of any information with respect to our inferential goals, we can reorganise the data into a collection of entries of the form

$$Y = (A, B, T_A, T_B).$$

That is, each entry  $Y$  contains a pair of successive stations a passenger arrives,  $A$  and  $B$ , with the times of arrival,  $T_A$  and  $T_B$ . Since the behaviour of all passengers will be treated as the same in this work, we do not keep the passenger ID in  $Y$ . If the original data are organised into  $M$  such entries, then the whole data set can be expressed as

$$\mathcal{Y} = \{Y_i = (A_i, B_i, T_{A,i}, T_{B,i}, i = 1, \dots, M)\}.$$

The inference problem is, then, to estimate the static parameters of the system, which are the  $\rho$  matrix and  $\alpha$ . In Section 2.3, the proposed method of estimation is explained.

## 2.3. Methodology

We propose to use the Metropolis-Hastings within Gibbs Sampling method in order to estimate the  $\rho$  matrix and  $\alpha$ . If we let  $y = (a, b, t_A, t_B)$  be the data portion that describes information for two consecutive entries of a passenger and  $\theta = (\alpha, \rho)$ ,



we have

$$p(y|\theta) \propto \sum_{d=1}^n \rho_{a,d} g_{\alpha}(b|d, \tau)$$

where the proportionality constant does not depend on  $\theta$ . For any  $y = (a, b, t_A, t_B)$  and  $\theta$ , let  $f(y|\theta)$  be defined as:

$$f(y|\theta) = \sum_{d=1}^n \rho_{a,d} g_{\alpha}(b|d, \tau) \quad (2.5)$$

where  $t_D$  is calculated from  $t_A$ ,  $a$ , and  $d$  using (2.1). Assume that there are  $M$  such entries  $y_{1:M} = y_1, \dots, y_M$ , so we have

$$y_{1:M} = \{A_i, B_i, t_{A,i}, t_{B,i}; i = 1, \dots, M\}.$$

Then, the likelihood of the whole data set can be expressed as:

$$p(y_{1:M}|\theta) \propto \prod_{m=1}^M f(y_m|\theta) \quad (2.6)$$

and the proportionality constant does not depend on  $\theta$ . In other words, the exact expression for  $p(y_{1:M}|\theta)$  contains additional multiplicative factors that stand for the probability density of the first arrival time of a passenger and the times of the next arrival, however, this density does not depend either on  $\rho$  or on  $\alpha$ , therefore, we can omit those factors.

Our aim is to estimate  $\theta$  in a Bayesian inference framework. Let the prior distribution for  $\theta$  has the density  $\mu(\theta)$ , which leads to the posterior distribution  $\pi(\theta|y_{1:M})$  that can be written as

$$\pi(\theta|y_{1:M}) \propto \mu(\theta)p(y_{1:M}|\theta). \quad (2.7)$$

For simplicity, we introduce a new variable,  $\eta = \log \alpha$ . The parameters  $\eta$  and  $\rho$  are a

*priori* independent with

$$p(\eta; \mu, \sigma^2) = \mathcal{N}(\eta; \mu, \sigma^2)$$

where  $\mathcal{N}(\eta; \mu, \sigma^2)$  is the density of the normal distribution with mean and variance parameters  $\mu$  and  $\sigma^2$ , and

$$\begin{aligned} p(\rho; \delta) &= \prod_{i=1}^n p(\rho_i; \delta_i) \\ &= \prod_{i=1}^n \text{Dirichlet}(\rho_{i,1}, \dots, \rho_{i,n}; \delta_{1,1}, \dots, \delta_{n,n}) \end{aligned}$$

where  $\text{Dirichlet}(\rho_{i,1}, \dots, \rho_{i,n}; \delta_{1,1}, \dots, \delta_{n,n})$  is the density of the Dirichlet distribution with parameters  $\delta_{i,1}, \dots, \delta_{i,n}$ . By design, we choose  $\delta_{i,i} = 0$  allowing no self transition among the stations. The pdf of the Gaussian  $(\mu, \sigma^2)$  is given by

$$\mathcal{N}(x; \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(x - \mu)^2\right).$$

Note that since the prior density of  $\eta$  was chosen to be a Gaussian density, the prior density of  $\alpha$  is then a Lognormal density with mean and variance parameters  $\mu$  and  $\sigma^2$  as  $\eta = \log \alpha$ . The pdf of  $\text{Dirichlet}(\delta_1, \dots, \delta_n)$  is given by

$$\text{Dirichlet}(x_1, \dots, x_n; \delta_1, \dots, \delta_n) = \left[ \frac{\prod_{i=1}^n \Gamma(\delta_i)}{\Gamma(\sum_{i=1}^n \delta_i)} \prod_{i=1}^n x_i^{\delta_i-1} \right] \mathbb{I}(x_1 + \dots + x_n = 1).$$

Overall, the prior density for  $\theta = (\rho, \eta)$  is

$$\mu(\theta) = p(\eta; \mu, \sigma^2) \prod_{i=1}^n p(\rho_i; \delta_i).$$

### 2.3.1. Markov chain Monte Carlo

It is generally hard to evaluate certain expectations with respect to their posterior distribution in (2.7). Therefore, we will use Markov chain Monte Carlo (MCMC). We extend the unknowns from  $\theta$  to  $\theta, D^{(1:M)}$  for the Metropolis Hastings within Gibbs algorithm since the data do not contain the departure information.

First of all,  $\theta$  is initialized, and in each iteration  $k$  of the Metropolis within Gibbs algorithm, the steps given in Algorithm 4 are conducted.

---

**Algorithm 4:**  $k$ 'th iteration of the Metropolis within Gibbs Sampling Algorithm

---

- 1 Sample  $D_{1:M}^{(k)} \sim p(D_{1:M}|\mathcal{Y}, \rho^{(k)}, \alpha^{(k)}) = \prod_{m=1}^M p(D_m|A_m, B_m, t_A, t_B, \rho^{(k)}, \alpha^{(k)})$
  - 2 Sample  $\rho^{(k)} \sim p(\rho|D_{1:M}^{(k)}, A_{1:M}, B_{1:M})$
  - 3 Run a one step Metropolis-Hastings (MH) algorithm for  $p(\alpha|D_{1:M}^{(k)}, A_{1:M}, B_{1:M}, t_{A,1:M}, t_{B,1:M})$  that updates  $\alpha^{(k-1)}$  to  $\alpha^{(k)}$
- 

The detailed explanation of the each step of the algorithm is as follows:

**2.3.1.1. Sampling  $D_{1:M}$ .** Since we do not have information on the departures, we first need to sample  $D$  for each entry  $i \in \{1, \dots, M\}$ . We calculate the likelihood of each  $d \in \{1, \dots, n\}$ ,  $p(D^{(1:M)}|\mathcal{Y}, \rho^{(k)}, \alpha^{(k)})$  and sample  $D$  from this likelihood. In order to calculate this likelihood, we use the above mentioned probability model. In order to do so, for each entry, we first calculate what the departure time would be for each  $d \in \{1, \dots, n\}$ . Since we know the next arrival station, we can then calculate what the time between departure and next arrival would be for each  $d \in \{1, \dots, n\}$ .

$$\mathbb{P}(D_i = d|A_i = a, B_i = b, T_{A,i} = t_A, T_{B,i} = t_B) \propto \rho_{a,d} g(b|d, \tau) \quad (2.8)$$

where  $t_D = t_A + |d - a|\Delta$ . Using this information about the time and the probability model mentioned above, we can then calculate the likelihood of departing at each  $d \in \{1, \dots, n\}$  and we can sample  $D$  for each entry.

2.3.1.2. Sampling  $\rho$  matrix. The information about the arrival stations and the sampled departure stations of each entry results can be used to construct an  $H$  matrix where  $(i, j)$ 'th element of the  $H$  matrix,  $h_{i,j}$ , denotes the total number of journeys that started at station  $i$  and ended at station  $j$ .  $h_{i,j}$  can be expressed as:

$$h_{i,j} = \sum_{m=1}^M \mathbb{I}(A_m = i, B_m = j). \quad (2.9)$$

The rows of the  $H$  matrix are distributed with Multinomial distribution with parameters being the corresponding row of the  $\rho$  matrix. Therefore, conditional on  $D$  and  $a$ , the rows of  $\rho$  are independent, and their conditional distribution depends only on  $D$ , from which  $H$  can be calculated. The Dirichlet distribution is a conjugate prior for the Multinomial distribution, hence for each row, this conditional distribution of  $(\rho_{i,1}, \dots, \rho_{i,n})$  is also a Dirichlet distribution. We can show this conditional density as

$$p(\rho_{i,1}, \dots, \rho_{i,n} | H) = \prod_{i=1}^n \text{Dirichlet}(\delta_{1,i} + h_{1,i}, \dots, \delta_{n,i} + h_{n,i}).$$

We can then use this posterior density to draw a sample for  $\rho$  since we can deduce that  $p(\rho | D_{1:M}, A_{1:M}, B_{1:M}) = p(\rho | H)$ .

2.3.1.3. Updating  $\alpha$ . Since the conditional probability of  $\alpha$  given other variables is not tractable, one step of the Metropolis Hastings algorithm within Gibbs Sampling is utilized since we can calculate the likelihood  $p(\alpha | D_{1:M}^{(k)}, A_{1:M}, B_{1:M}, t_{A,1:M}, t_{B,1:M})$  up to a proportionality constant. The acceptance probability of the MH move becomes:

$$\min \left\{ 1, \frac{p(\alpha' | D_{1:M}^{(k)}, A_{1:M}, B_{1:M}, t_{A,1:M}, t_{B,1:M}) q(\alpha | \alpha')}{p(\alpha | D_{1:M}^{(k)}, A_{1:M}, B_{1:M}, t_{A,1:M}, t_{B,1:M}) q(\alpha' | \alpha)} \right\}.$$

As mentioned previously, we defined a new variable,  $\eta = \log \alpha$ , whose prior density is a Gaussian. This conversion assures the positivity of the value of  $\alpha$  and also results in more convenient calculations, therefore  $\eta$  was proposed and updated and the value of  $\alpha$  was calculated thereon. We decided to use a symmetric random walk for our proposal density, which is Gaussian,  $q(\eta|\eta') = q(\eta'|\eta)$ . The acceptance probability can then be expressed as:

$$\min \left\{ 1, \frac{p(\eta'|D_{1:M}^{(k)}, A_{1:M}, B_{1:M}, t_{A,1:M}, t_{B,1:M})}{p(\eta|D_{1:M}^{(k)}, A_{1:M}, B_{1:M}, t_{A,1:M}, t_{B,1:M})} \right\}. \quad (2.10)$$

We cannot directly compute the exact value of  $p(\eta|D_{1:M}^{(k)}, A_{1:M}, B_{1:M}, t_{A,1:M}, t_{B,1:M})$ , but since the value of this likelihood is proportional to the joint density

$$p(\eta, D_{1:M}^{(k)}, A_{1:M}, B_{1:M}, t_{A,1:M}, t_{B,1:M}),$$

we can compute its value up to a proportionality constant.

$$p(\eta, D_{1:M}^{(k)}, A_{1:M}, B_{1:M}, t_{A,1:M}, t_{B,1:M}) = p(\eta)p(\rho) \prod_{m=1}^M p(A_m)\rho_{A_m, D_m} g_\alpha(B_m|D_m, \tau) \quad (2.11)$$

If we eliminate the terms which do not depend on  $\eta$ , we can conclude that the likelihood probability is proportional to the term:

$$p(\eta) \propto \prod_{i=1}^M g_\alpha(B_m|D_m, \tau).$$

With the help of these derivations, we calculate the acceptance probability to be:

$$\min \left\{ 1, \frac{p(\eta') \prod_{i=1}^M g_{\alpha'}(B_m|D_m, \tau)}{p(\eta) \prod_{i=1}^M g_\alpha(B_m|D_m, \tau)} \right\}. \quad (2.12)$$

As mentioned above, the prior density for  $\eta$  was chosen to be a Gaussian density; hence, we can calculate the ratio given in 2.12.

## 2.4. Results

In the absence of real data from the relevant government office; the data were synthetically generated using MATLAB with compliance with the missing data model. The behavior of the passengers was simulated for a period of 72 hours until there was a total of  $M$ , in our case 100000, entries. The generated data contains:

- The  $\rho$  matrix,
- Passenger ID,
- The station at which the passenger arrives,  $A \in \{1, \dots, n\}$ ,
- The time of arrival,  $T_A$ ,
- The station from which the passenger leaves,  $D \in \{1, \dots, n\}$ ,
- The time of departure,  $T_D$ ,
- The station at which the passenger arrives next,  $B \in \{1, \dots, n\}$ .
- The time of the next arrival,  $T_B$ .

We assume that the time between departure and the next arrival is exponentially distributed with a mean of 240 minutes. We have simulated for  $n = 10$  stations. The generation of the data simulates such behavior that a passenger enters the system from a random station at a random time. The departure station is then sampled from  $\rho$ . The time spent between the departure and the next arrival is then sampled from exponential distribution with a mean of 240. The probabilities of each station being the next arrival station are calculated, then the next arrival is sampled. After generating the data, we modify it by removing the information regarding the departure stations and the departure times since the data we would have received would not contain this information. However, since we have generated the data, we know the true values of  $\alpha$  and the elements of the  $\rho$  matrix which can later be used to check the validity and consistency of the algorithm.

We ran the Metropolis within Gibbs Algorithm for 20000 iterations for different values of  $\alpha$ . We have compared the true values with our estimations. The values of  $\alpha$  and the elements of the  $\rho$  matrix through 20000 iterations compared to the true value

for  $\alpha = 0.001$  can be seen in Figure 2.1 and Figure 2.2.

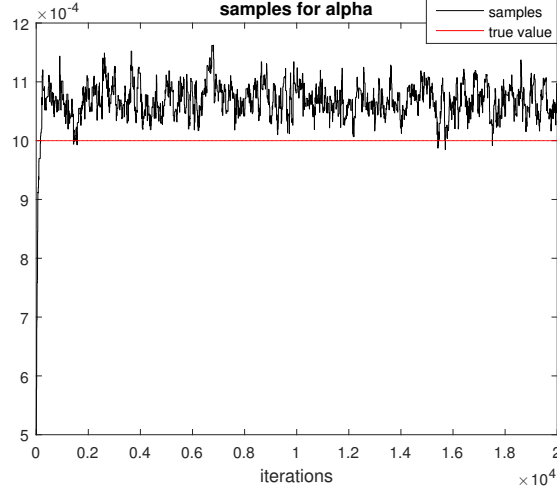


Figure 2.1. Comparison of the estimated  $\alpha$  with its true value through iterations when  $\alpha = 0.001$

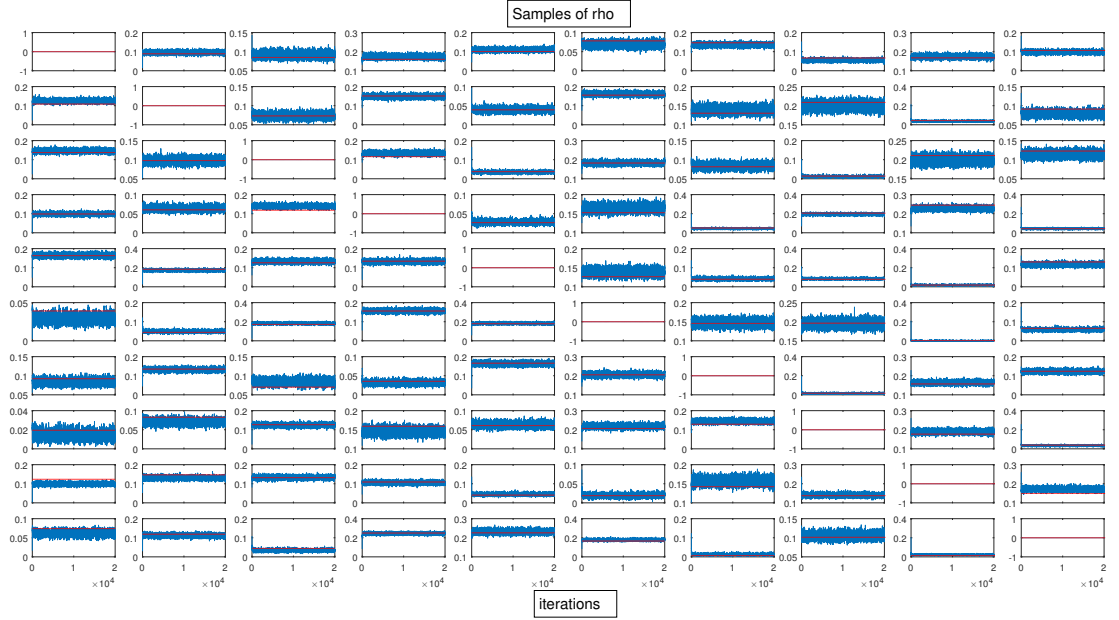


Figure 2.2. Comparison of the estimated  $\rho$  with their true values for  $\alpha = 0.001$

In this configuration, 46814 passengers out of 100000 returned to the station of departure for their next arrival. We have also tested our model with higher and lower rates of returning to the departure station. For example, if  $\alpha$  is reduced, the probability of returning to the same station will increase and vice-versa. The values of  $\alpha$  and the elements of the  $\rho$  matrix through 20000 iterations compared to the true values for  $\alpha = 0.0001$  can be seen in Figure 2.3 and Figure 2.4.

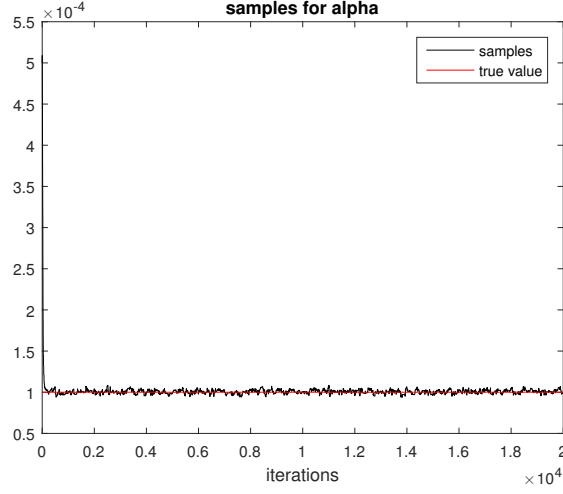


Figure 2.3. Comparison of the estimated  $\alpha$  with its true value through iterations  
when  $\alpha = 0.0001$

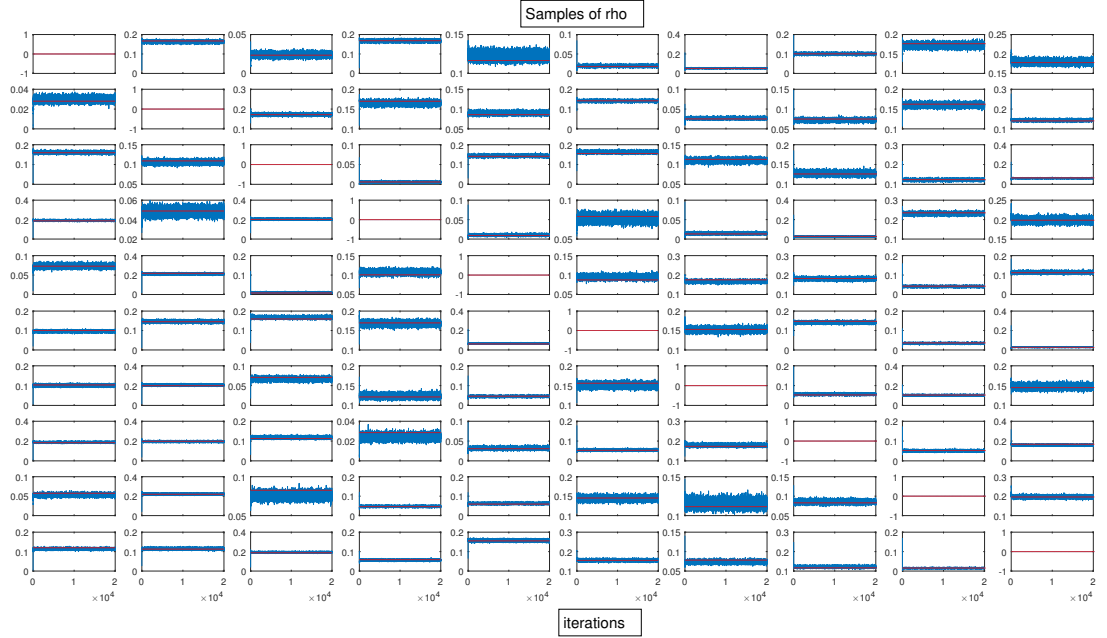


Figure 2.4. Comparison of the estimated  $\rho$  with their true values for  $\alpha = 0.0001$

In this configuration 84476 passengers out of 100000 returned to the station of departure for their next arrival. This increase in the number of passengers returning to the station of departure is due to the decrease in the parameter  $\alpha$  which causes passengers to choose their next entering station less randomly.

Similarly, increase in  $\alpha$  will cause passengers to choose their next entering station more randomly and therefore more homogeneously. The value of  $\alpha$  was then set to



0.01 in order to simulate this phenomenon and conveniently the number of passengers returning to the station of departure is 17922. The results regarding this configuration can be seen in Figure 2.5 and Figure 2.6.

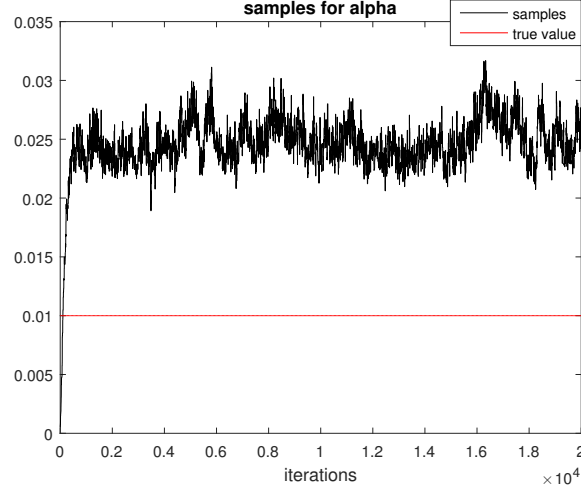


Figure 2.5. Comparison of the estimated  $\alpha$  with its true value through iterations  
when  $\alpha = 0.01$

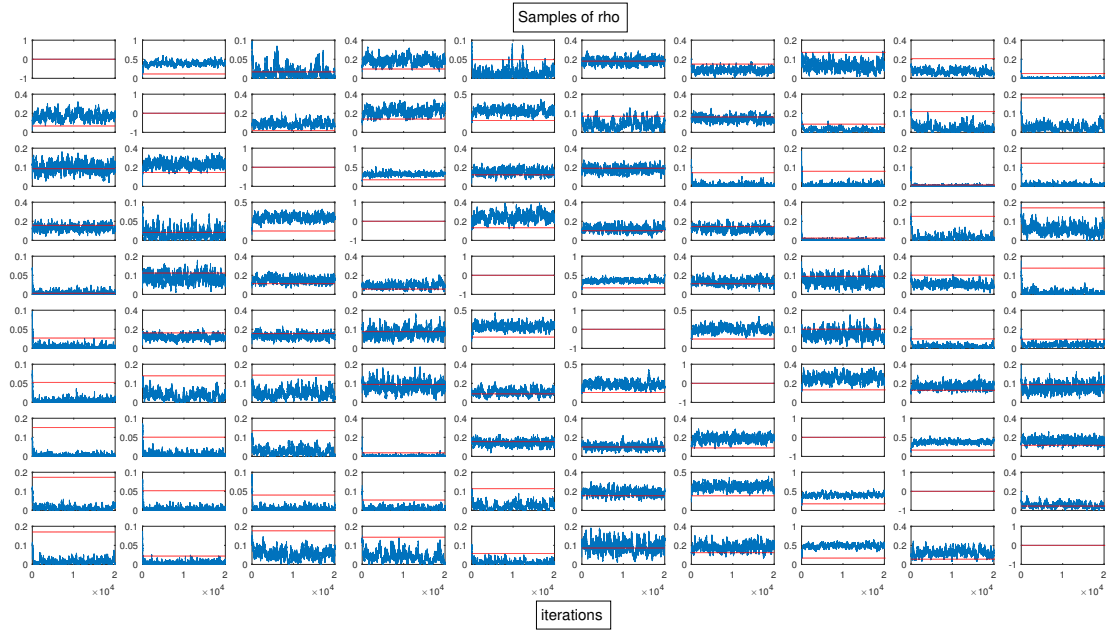


Figure 2.6. Comparison of the estimated  $\rho$  with their true values for  $\alpha = 0.01$

## 2.5. Discussion

In order to measure the performance of the algorithm, we have calculated two different MSE's,  $MSE_1$  and  $MSE_2$ . How these values are calculated are as follows:

$$MSE_1 = \frac{1}{T - t_{\text{burn}}} \sum_{t=t_{\text{burn}}+1}^T \sum_{i=1}^n \sum_{j=1}^n (\rho_{i,j}^{(t)} - \rho_{i,j}^*)^2$$

$$MSE_2 = \sum_{i=1}^n \sum_{j=1}^n \left[ \frac{1}{T - t_{\text{burn}}} \sum_{t=t_{\text{burn}}+1}^T \rho_{i,j}^{(t)} - \rho_{i,j}^* \right]^2$$

where  $T$  denotes the total number iterations and  $t_{\text{burn}}$  denotes the number of iterations in the burn-in period. Moreover,  $\rho^*$  denotes the true posterior mean of  $\rho$  given the the true  $H$  matrix. As mentioned previously, the posterior density  $p(\rho|H)$  is a Dirichlet distribution and since the the true values of  $\rho$  and  $H$  matrices are stored after the data generation, the true posterior mean can be calculated. Table 2.2 and Figure 2.7 shows the MSE values obtained for several runs under different  $\alpha$  values.

Table 2.2.  $MSE_1$  and  $MSE_2$  values for different values of  $\alpha$

$\alpha$	$MSE_1$	$MSE_2$
0.0001	$1.6766 \times 10^{-4}$	$5.1603 \times 10^{-5}$
0.0002	$2.4188 \times 10^{-4}$	$1.0312 \times 10^{-4}$
0.0005	$5.7937 \times 10^{-4}$	$4.0539 \times 10^{-4}$
0.001	0.0013	0.0011
0.002	0.0026	0.0022
0.005	0.0218	0.0208
0.01	0.192	0.1893
0.02	0.2446	0.2407
0.05	0.4305	0.4259

As it can be seen at Table 2.2 and Figure 2.7, both of the MSE values tend to increase as the  $\log \alpha$ , hence  $\alpha$  value increases.

This result was expected since as  $\alpha$  increases, the effect of the time passed decreases and passengers start to move in the system more randomly and choose their

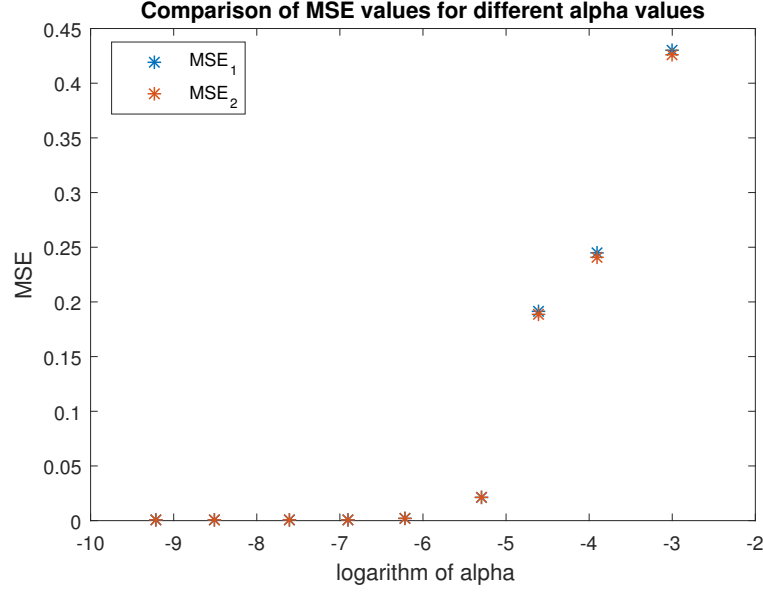


Figure 2.7. Comparison of the MSE values for different  $\log \alpha$  levels

next station to enter the metro-line more homogeneously. This phenomenon causes the algorithm to estimate the  $\rho$  matrix of the configuration with a lower  $\alpha$  value more accurately with its MSE values being quite low,  $1.6766 \times 10^{-4}$  and  $5.1603 \times 10^{-5}$ , whereas the algorithm estimates  $\rho$  matrix of the configuration with higher  $\alpha$  value less accurately with its MSE values being relatively larger, 0.192 and 0.1893. Nevertheless, these results prove the algorithm to be valid when estimating the  $\rho$  matrix of such a scenario with missing data models. Ni and Leonard II (2005) showed in their work that Markov chain Monte Carlo methods can be used to simulate the missing data for traffic counts, speed, and density. We have expanded this approach to simulate the missing destination data and further to estimate the origin-destination matrix of Istanbul metro-line.

### 3. BAYESIAN ODM PARAMETER ESTIMATION USING THE NOISY DATA

#### 3.1. The Available Data and the Model

As mentioned in Chapter 1, another problem one might come across when estimating the origin-destination matrix is that the data provided by the data holders might not be the original data collected directly from the passengers. Due to data privacy issues and in order to protect each passengers private travel information, the data holder is highly likely to provide a noisy version of the original data. In Chapter 2, we have defined  $H$  matrix to be an  $n \times n$  ODM of a single line railway system with  $n$  many stations where  $h_{i,j}$  denotes the total number of journeys taken from station  $i$  to station  $j$ . In a scenario where the whole data is collected, we can assume that each user's journeys are recorded and each user has their own  $n \times n$   $H$  matrix denoted as  $H_u$ . Similar to  $H$ ,  $(i, j)$ 'th element of each of the  $H_u$ 's denotes the total number of journeys made from station  $i$  to station  $j$  by the user. The following table is an example of an  $H_u$  where  $n = 3$ :

Table 3.1. An example  $H_u$  where  $n = 3$

0	2	1
3	0	2
1	1	0

With this information we can consider the  $H$  matrix to be the summation of  $H$  matrices regarding each user individually,  $H_u$ . Let  $U$  be the total number of users, then

$$H = \sum_{u=1}^U H_u \quad (3.1)$$

For this model, it is assumed that the data holder will release a noisy version of  $H$  matrix denoted as  $\tilde{H}$  produced via the Laplace mechanism from the original  $H$

matrix. In order to formulate this inferential problem, it is essential to understand the mechanism that produces the noisy origin-destination matrices

$$(\tilde{H}_1, \dots, \tilde{H}_D)$$

for a total of  $D$  periods; for example each period may be a day long.  $\tilde{H}$ 's will be the only data available in order to estimate the  $\rho$  matrix of this system. It is assumed that, the passengers arrive to each station  $i$  according to a Poisson arrival process with parameters  $(\lambda_1, \dots, \lambda_n)$ . The  $\lambda_i$ 's are assumed to be independently and identically distributed from some Gamma distribution  $\text{Gamma}(\alpha, \beta)$ . Let  $X_{d,i}$  denote the number of total passengers arrived to the station  $i$  on day  $d$  due to this Poisson arrival process.

$$x_{d,i} \stackrel{\text{i.i.d.}}{\sim} \mathcal{PO}(\lambda_i), \quad d = 1, \dots, D.$$

By construction,  $X_{d,i}$  can be computed from the  $H$  matrix as follows:

$$x_{d,i} = \sum_{j=1}^n h_{d,i,j}, \quad i = 1, \dots, n. \quad (3.2)$$

Similarly, if we let  $\tilde{X}_i$  denote the number of passengers arrived to station  $i$  according to the noisy data, then  $\tilde{X}_i$ 's can be computed through the  $\tilde{H}$  matrix.

When a passenger arrives to station  $i$ , the probability of him departing to a station  $j, j \neq i$ , is denoted as  $\rho_{i,j}$ . Our prior belief is that each row of the  $\rho$  matrix is distributed with some Dirichlet distribution similar to Chapter 2. Passengers arrive to the system as a result of a Poisson arrival process and depart to stations according to probability matrix  $\rho$  and through this process, the  $H$  matrix is produced. By this

mechanism, it can be seen that the joint density of the  $H$  matrixes given  $\rho$  are

$$\begin{aligned} p(H; X, \rho) &= \prod_{d=1}^D \prod_{i=1}^n p(h_{d,i,:}; x_{d,i}, \rho_i) \\ &= \prod_{d=1}^D \prod_{i=1}^n \text{Multinomial}(h_{d,i,:}; x_{d,i}, \rho_i) \end{aligned}$$

where  $h_{i,j}$  refers to the  $(i, j)$ 'th element of the  $H$  matrix. The pmf of the Multinomial distribution is given by

$$\text{Multinomial}(x_1, \dots, x_k; n, \pi_1, \dots, \pi_k) = \frac{n!}{x_1! \dots x_k!} \pi_1^{x_1} \dots \pi_k^{x_k}.$$

As mentioned above, the data holder will not release this original  $H$  matrix but will add a Laplacian noise distributed with  $\text{Laplace}(\frac{S}{\epsilon})$ .  $S$  is the sensitivity of the data and defined to be

$$S = \max_{y, y': h(y, y') \leq 1} h_{d,i,j} - h'_{d,i,j}. \quad (3.3)$$

This in fact is the maximum difference between two adjacent datasets  $y$  and  $y'$  differing in one user's information. It is assumed that the data is collected for five days and each passenger can make the same journey from a specific station  $i$  to station  $j$  for a maximum of two times, therefore the sensitivity of this model,  $S$  is equal to 10. With the addition of this Laplacian noise, the density of each  $\tilde{H}$  is

$$p(\tilde{H}; H, S/\epsilon) = \prod_{i=1}^n \prod_{j=1}^n \text{Laplace}(\tilde{h}_{d,i,j}; h_{d,i,j}, S/\epsilon).$$

The pdf of the Laplace distribution is given by

$$p(x; \mu, b) = \frac{1}{2b} \exp\left(\frac{-|x - \mu|}{b}\right).$$

For this model we will assume that  $H_N$  is the only data available in use to conduct

statistical analysis and estimate the  $\rho$  matrix.

### 3.2. The Inferential Problem

The organisation of the available data is as follows:

$$y = (\tilde{h}, \tilde{x})$$

where  $\tilde{h}$  denotes one row of the  $\tilde{H}$  and  $\tilde{x}$  denotes the sum of entries in this row. The data is assumed to be organised into  $n$  such entries, then the whole data set can be expressed as:

$$Y = \left\{ y_{d,i} = (\tilde{h}_{d,i,:}, \tilde{x}_{d,i}), i = 1, \dots, n, d = 1, \dots, D \right\}$$

where  $Y$  is the noisy  $\tilde{H}$  matrixes obtained for  $D$  many days and the  $\tilde{x}_{d,i}$ 's computed through this matrix. The inferential problem regarding this model then becomes estimating the static parameter of this system  $\rho$ , and according to the hierarchical structure described in the previous section,  $\lambda$ . Our main goal does not include estimating  $\lambda$ , however we need this estimation in order to estimate  $\rho$ .

### 3.3. Methodology

Our proposition to solve this estimation problem is to use Metropolis within Gibbs algorithm as we proposed for the previous model. If we let  $y = (\tilde{h}, \tilde{x})$  be the one row of the noisy origin-destination matrix obtained from the data holder, and  $\theta = (\rho, \lambda, x, h)$  where  $x, h, \rho, \lambda$  denotes number of customers arrived to a specific station in five days and the row of the  $H$  matrix associated with this station, row of the  $\rho$  matrix associated with this station, and  $\lambda$  associated with this station respectively.

The likelihood of the whole data can be expressed as:

$$p(y_{1:n}|\theta) \propto \prod_{i=1}^n f(y_i|\theta),$$

and similarly to the previous model, the proportionality constant does not depend on  $\theta$ . We are again working in a Bayesian inference framework, therefore the prior and the posterior of  $\theta$  and  $\theta$  given the data need to be defined. Let  $\mu(\theta)$  denote the prior distribution for  $\theta$  and  $\pi(\theta|y_{1:n})$  denote the posterior distribution derived from this prior distribution. We can say:

$$\pi(\theta|y_{1:n}) \propto \mu(\theta)p(y_{1:n}|\theta).$$

The parameters  $\rho$  and  $\lambda$  are *a priori* independent with

$$\begin{aligned} p(\lambda; \alpha, \beta) &= \prod_{i=1}^n p(\lambda_i; \alpha, \beta) \\ &= \prod_{i=1}^n \text{Gamma}(\lambda_i; \alpha, \beta) \end{aligned}$$

where  $\text{Gamma}(\lambda_i; \alpha, \beta)$  is the density of the gamma distribution with shape and scale parameters  $\alpha$  and  $\beta$ , and

$$\begin{aligned} p(\rho; \delta) &= \prod_{i=1}^n p(\rho_i; \delta_i) \\ &= \prod_{i=1}^n \text{Dirichlet}(\rho_{i,1}, \dots, \rho_{i,n}; \delta_{1,1}, \dots, \delta_{n,n}) \end{aligned}$$

where  $\text{Dirichlet}(\rho_{i,1}, \dots, \rho_{i,n}; \delta_{1,1}, \dots, \delta_{n,n})$  is the density of the Dirichlet distribution with parameters  $\delta_{i,1}, \dots, \delta_{i,n}$ . By design, we choose  $\delta_{i,i} = 0$  allowing no self transition among the stations. The pdf of the  $\text{Gamma}(\alpha, \beta)$  is given by

$$\text{Gamma}(x; \alpha, \beta) = \frac{\beta^\alpha x^{\alpha-1} e^{-\beta x}}{\Gamma(\alpha)}.$$



The pdf of  $\text{Dirichlet}(\delta_1, \dots, \delta_n)$  is given by

$$\text{Dirichlet}(x_1, \dots, x_n; \delta_1, \dots, \delta_n) = \left[ \frac{\prod_{i=1}^n \Gamma(\delta_i)}{\Gamma(\sum_{i=1}^n \delta_i)} \prod_{i=1}^n x_i^{\delta_i-1} \right] \mathbb{I}(x_1 + \dots + x_n = 1).$$

Let

$$h_{d,i} = (H_d(i, 1), \dots, H_d(i, n)), \quad \tilde{h}_{d,i} = (\tilde{H}_d(i, 1), \dots, \tilde{H}_d(i, n))$$

be the  $i$ 'th row of  $H_d$  and  $\tilde{H}_d$ , respectively. The joint distribution of  $\theta = (\rho, \lambda, X_{1:D}, H_{1:D})$  and  $y = \{\tilde{H}_d, \quad d = 1, \dots, D\}$  is

$$p(\theta, y) = \prod_{i=1}^n \left\{ p(\lambda_i; \alpha, \beta) p(\rho_{i,:}; \delta_{i,:}) \prod_{d=1}^D \left[ p(x_{d,i} | \lambda_i) p(h_{d,i,:} | x_{d,i}, \rho_{i,:}) p(\tilde{h}_{d,i,:} | h_{d,i,:}) \right] \right\}. \quad (3.4)$$

We can express the the density  $p(X; \lambda)$  as

$$\begin{aligned} p(X; \lambda) &= \prod_{i=1}^n p(x_{d,i} | \lambda_i) \\ &= \prod_{i=1}^n \text{Poisson}(x_{d,i}; \lambda_i). \end{aligned}$$

The pmf of the Poisson distribution is given by

$$p(x; \lambda) = \frac{e^{-\lambda} \lambda^x}{x!} \text{ for } x = 0, 1, 2, \dots$$

Since  $p(\theta, y)$  can fully be factorised over the variables regarding the rows of the ODM, we can target the posterior of the variables corresponding to each row separately. In other words, a separate instance of the same MCMC algorithm can be used for each row.

### 3.3.1. Markov Chain Monte Carlo - Type 1

Our proposed method of solving this estimation problem is again using Metropolis within Gibbs algorithm which starts by initializing

$$\theta^{(1)} = (\rho^{(1)}, \lambda^{(1)}, x_1^{(1)}, \dots, x_D^{(1)}, h_1^{(1)}, \dots, h_D^{(1)})$$

and the algorithm for this type of proposal at iteration  $k$  is given at Algorithm 5.

---

**Algorithm 5:** Metropolis within Gibbs Sampling Algorithm at iteration  $k$

---

```

1 for  $i = 1, \dots, n$  do
2   for  $d = 1, \dots, D$  do
3     Update  $(x_{d,i}^{(k)}, h_{d,i,:}^{(k)})$  to  $(x_{d,i}^{(k+1)}, h_{d,i,:}^{(k+1)})$  by a MH move that targets
4        $p(x'_{d,i}, h'_{d,i,:} | \tilde{h}_{d,i,:}, \lambda_i^{(k)}, \rho_{i,:}^{(k)})$ 
5     Sample  $\lambda_i^{(k+1)} \sim p(\lambda_i | x_{1,i}^{(k+1)}, \dots, x_{D,i}^{(k+1)})$ 
6     Sample  $\rho_{i,:}^{(k+1)} \sim p(\rho_{i,:} | h_{1,i,:}^{(k+1)}, \dots, h_{D,i,:}^{(k+1)})$ 

```

---

3.3.1.1. Updating  $(x_{1,i}, \dots, x_{D,i})$  and  $(h_{1,i,:}, \dots, h_{D,i,:})$ . Computing the full conditional probability of these parameters are not computationally feasible. In order to overcome this problem, we propose to update these parameters with an Metropolis-Hastings move that targets

$$p(x'_{1,i}, \dots, x'_{D,i}, h'_{1,i,:}, \dots, h'_{D,i,:} | \tilde{h}_{1,i,:}, \dots, \tilde{h}_{D,i,:}, \lambda_i, \rho_i).$$

As mentioned previously, we can conduct the steps for each  $x_{d,i}$  and  $h_{d,i,:}$ . We can express this probability as:

$$\begin{aligned} p(x_{d,i} | \tilde{h}_{d,i,:}, \lambda_i, \rho_{i,:}) &= p(\rho_{i,:})p(\lambda_i)p(x_{d,i} | \lambda_i)p(h_{d,i,:} | x_i, \lambda_i)p(\tilde{h}_{d,i,:} | h_{d,i,:}) \\ &\propto p(x_{d,i} | \lambda_i)p(h_{d,i,:} | x_{d,i}, \lambda_i)p(\tilde{h}_{d,i,:} | h_{d,i,:}). \end{aligned}$$

We have decided to propose the new values for  $(x_{1,i}, \dots, x_{D,i})$  and  $(h_{1,i}, \dots, h_{D,i})$  from the prior distribution of these parameters given the current values of the other parameters. Each  $x_{d,i}$  and hence  $h_{d,i,:}$  are proposed and updated individually. Therefore we will use the following proposal density for each  $x_{d,i}$ :

$$q(x'_{d,i} h'_{d,i,:} | x_{d,i}, h_{d,i,:}, \rho_i, \lambda_i) = p(x'_{d,i} | \lambda_i) p(h'_{d,i,:} | x'_{d,i}, \lambda_i). \quad (3.5)$$

The acceptance probability of this MCMC kernel then becomes:

$$\min \left\{ 1, \frac{p(x'_{d,i} | \lambda_i) p(h'_{d,i,:} | x'_{d,i}, \lambda_i, \rho_i) p(x_{d,i} | \lambda_i) p(h_{d,i,:} | x_{d,i}, \lambda_i, \rho_i) p(\tilde{h}_{d,i,:} | h'_{d,i,:})}{p(x_{d,i} | \lambda_i) p(h_{d,i,:} | x_{d,i}, \lambda_i, \rho_i) p(x'_{d,i} | \lambda_i) p(h'_{d,i,:} | x'_{d,i}, \lambda_i, \rho_i) p(\tilde{h}_{d,i,:} | h_{d,i,:})} \right\}.$$

Which then simplifies to:

$$\min \left\{ 1, \frac{p(\tilde{h}_{d,i,:} | h'_{d,i,:})}{p(\tilde{h}_{d,i,:} | h_{d,i,:})} \right\}. \quad (3.6)$$

Conveniently, the probability  $p(\tilde{h}_{d,i,:} | h_{d,i,:})$  can be calculated easily since the density  $p(\tilde{h}_{d,i,:} | h_{d,i,:})$  is  $\text{Laplace}(h_{d,i,j}, \frac{S}{\epsilon})$ . We can therefore express this acceptance probability as:

$$\min \left\{ 1, \frac{\frac{\epsilon}{2S} \exp \left( \frac{\sum_{j=1}^n -\epsilon |\tilde{h}_{d,i,j} - h'_{d,i,j}|}{2S} \right)}{\frac{\epsilon}{2S} \exp \left( \frac{\sum_{j=1}^n -\epsilon |\tilde{h}_{d,i,j} - h_{d,i,j}|}{2S} \right)} \right\}.$$

This probability can be simplified as:

$$\min \left\{ 1, \exp \left( \frac{-\epsilon \sum_{j=1}^n \left[ |\tilde{h}_{d,i,j} - h'_{d,i,j}| + |\tilde{h}_{d,i,j} - h_{d,i,j}| \right]}{2S} \right) \right\}. \quad (3.7)$$

3.3.1.2. Sampling  $\lambda_i$ . In this step,  $\lambda_i$  is sampled from  $p(\lambda_i|x_{1,i}, \dots, x_{D,i})$  for all  $i = 1, \dots, n$ . We can express this probability as:

$$p(\lambda_i|x_{1,i}, \dots, x_{D,i}) = p(\lambda_i)p(x_{1,i}|\lambda_i) \dots p(x_{D,i}|\lambda_i). \quad (3.8)$$

$p(\lambda_i)$  is the pdf of the Gamma distribution evaluated at the current  $\lambda$  values with parameters,  $\alpha$  and  $\beta$  and  $p(x_{d,i}|\lambda_i)$  is the pdf of the Poisson distribution evaluated at  $x_i$  with parameters  $\lambda_i$ . Since Gamma distribution is the conjugate prior for the Poisson distribution, this posterior density is also a pdf of a Gamma distribution. We can show this mechanism as follows:

$$p(\lambda_i|x_{1,i}, \dots, x_{D,i}) \propto \frac{\beta^\alpha}{\Gamma(\alpha)} \lambda_i^{\alpha-1} e^{-\beta\lambda_i} \prod_{d=1}^D \frac{\lambda_i^{x_{d,i}} e^{-\lambda_i}}{x_{d,i}!}.$$

We can see that:

$$p(\lambda_i|x_{1,i}, \dots, x_{D,i}) \propto \lambda_i^{\alpha + \sum_{d=1}^D x_{d,i} - 1} e^{-\lambda_i(\beta + D)}.$$

Therefore the posterior density  $p(\lambda_i|x_{1,i}, \dots, x_{D,i})$  is also a Gamma distribution with parameters  $(\alpha + \sum_{d=1}^D x_{d,i})$  and  $(\beta + D)$ .

3.3.1.3. Sampling  $\rho_{i,:}$ . In this step  $\rho_{i,:}$  is sampled from  $p(\rho_{i,:}|h_{1,i,:}, \dots, h_{D,i,:})$  for all  $i = 1, \dots, n$ . This probability can be expressed as:

$$p(\rho_{i,:}|h_{1,i,:}, \dots, h_{D,i,:}) = p(\rho_{i,:}) \prod_{d=1}^D p(h_{d,i,:}|\rho_{i,:}). \quad (3.9)$$

$p(\rho_{i,:})$  is the pdf of the Dirichlet distribution evaluated at the current  $\rho$  values with parameters,  $\delta_i, i = 1, \dots, n$  whereas  $p(h_{d,i,:}|\rho_{i,:})$  is the pdf of the Multinomial distribution evaluated at  $h_{d,i,:}$  with parameters  $n$  and  $x_{d,i}$ . Since the Dirichlet distribution is the conjugate prior for the Multinomial distribution, this posterior density is also a

Dirichlet distribution. We can evaluate this posterior density as:

$$p(\rho_{i,:} | h_{1,i,:}, \dots, h_{D,i,:}) \propto \frac{\prod_{j=1}^n \Gamma(\delta_{i,j})}{\Gamma(\sum_{j=1}^n \delta_{i,j})} \left( \prod_{j=1}^n \rho_{i,j}^{\delta_{i,j}-1} \right) \prod_{d=1}^D \left( \frac{x_{d,i}!}{h_{d,i,1}! \dots h_{d,i,n}!} \prod_{j=1}^n \rho_{i,j}^{h_{d,i,j}} \right). \quad (3.10)$$

Therefore:

$$p(\rho_{i,:} | h_{1,i,:}, \dots, h_{D,i,:}) \propto \prod_{j=1}^n \rho_{i,j}^{\delta_{i,j}-1+\sum_{d=1}^D h_{d,i,j}}.$$

We can deduce that the posterior density  $p(\rho_{i,:} | h_{1,i,:}, \dots, h_{D,i,:})$  is also a Dirichlet distribution with parameters  $(\delta_{i,:} + \sum_{d=1}^D h_{d,i,:})$ .

### 3.3.2. Markov Chain Monte Carlo - Type 2

In order to understand the effect of the proposal density on the performance of the algorithm, we have designed other proposal densities for the Metropolis-Hastings move. On of these proposal densities, namely *Type 2* proposes  $X$  from the Poisson  $(\frac{x_{d,i} + \lambda_i}{2})$ . To sum up, the  $k$ 'th iteration of this algorithm is given at Algorithm 6.

---

**Algorithm 6:** Metropolis within Gibbs Sampling Algorithm at iteration  $k$

---

```

1 for  $i = 1, \dots, n$  do
2   for  $d = 1, \dots, D$  do
3     Update  $(x_{d,i}^{(k)}, h_{d,i,:}^{(k)})$  to  $(x_{d,i}^{(k+1)}, h_{d,i,:}^{(k+1)})$  by a MH move that targets
        $p(x'_{d,i}, h'_{d,i,:} | x_{d,i}^{(k)}, \tilde{h}_{d,i,:}^{(k)}, \lambda_i^{(k)}, \rho_i^{(k)})$ 
4     Sample  $\lambda_i^{(k+1)} \sim p(\lambda_i | x_{1,i}^{(k+1)}, \dots, x_{D,i}^{(k+1)})$ 
5     Sample  $\rho_{i,:}^{(k+1)} \sim p(\rho_{i,:} | h_{1,i,:}^{(k+1)}, \dots, h_{D,i,:}^{(k+1)})$ 

```

---

Each  $x_{d,i}$  and hence  $h_{d,i,:}$  can be proposed and updated individually. In more detail, the proposal density can be expressed as:

$$q(x'_{d,i}, h_{d,i,:}, | x_{d,i} \lambda_i, \rho_{i,:}) = q(x'_{d,i} | x_{d,i}, \lambda_i) q(h'_{d,i,:} | x_{d,i}, \lambda_i, \rho_{i,:}). \quad (3.11)$$

When the proposal density is selected as this density, the acceptance probability of the Metropolis Hastings move step of the algorithm is changed to the following:

$$\min \left\{ 1, \frac{p(x'_{d,i}|\lambda_i)p(h'_{d,i,:}|x'_{d,i}, \lambda_i, \rho_{i,:})p(\tilde{h}_{d,i,:}|h'_{d,i,:})q(x_{d,i}|x_{d,i}, \lambda_i)q(h_{d,i,:}|x_{d,i}, \lambda_i, \rho_{i,:})}{p(x_{d,i}|\lambda_i)p(h_{d,i,:}|x_{d,i}, \lambda_i, \rho_{i,:})p(\tilde{h}_{d,i,:}|h_{d,i,:})q(x'_{d,i}|x_{d,i}, \lambda_i)q(h'_{d,i,:}|x'_{d,i}, \lambda_i, \rho_{i,:})} \right\}.$$

This probability can then be simplified to:

$$\min \left\{ 1, \frac{p(x'_{d,i}|\lambda_i)q(x_{d,i}|x_{d,i}, \lambda_i)p(\tilde{h}_{d,i,:}|h'_{d,i,:})}{p(x_{d,i}|\lambda_i)q(x'_{d,i}|x_{d,i}, \lambda_i)p(\tilde{h}_{d,i,:}|h_{d,i,:})} \right\}. \quad (3.12)$$

This expression can then be simplified to:

$$\min \left\{ 1, e^{\hat{\lambda}_i - \tilde{\lambda}'_i} \left( \frac{\lambda_i}{\tilde{\lambda}_i} \right)^{x'_{d,i} - x_{d,i}} \frac{p(\tilde{h}_{d,i,:}|h'_{d,i,:})}{p(\tilde{h}_{d,i,:}|h_{d,i,:})} \right\}. \quad (3.13)$$

By using the same calculation used in Type 1 for the term  $\frac{p(\tilde{h}_{d,i,:}|h'_{d,i,:})}{p(\tilde{h}_{d,i,:}|h_{d,i,:})}$ , this probability can be easily computed. The other steps of the algorithm for updating  $\lambda$  and  $\rho$  is the same with Type 1.

### 3.3.3. Markov Chain Monte Carlo - Type 3

Since we are aiming to estimate the  $X$  vector and the  $H$  matrix, we can design a proposal density which proposes  $(x_{1,1}, \dots, x_{D,i})$  and  $(h_{1,i,:}, \dots, h_{D,i,:})$  from their marginal densities. This type of proposal density allows us to run the Metropolis Hastings algorithm without the Gibbs sampling algorithm extension, and steps conducted at iteration  $k$  is given at Algorithm 7.

In more detail, this proposal density allows us to run the Metropolis Hastings algorithm without updating  $\lambda$  and  $\rho$  since for this type of proposal density  $\lambda$  and  $\rho$  are not updated, therefore not stored. It is possible to propose each  $x_{d,i}$  from its marginal

---

**Algorithm 7:** Metropolis Hastings Algorithm regarding Type 3 proposal density at iteration  $k$

---

```

1 for  $i = 1, \dots, n$  do
2   for  $d = 1, \dots, D$  do
3     Draw an  $x'_{d,i}$  from its marginal density  $\sim \int_{-\infty}^{\infty} p(x'_{d,i}|\lambda_i)p(\lambda_i)d\lambda_i$ 
4     Draw an  $h'_{d,i,:}$  from its marginal density
        $\sim \int_{-\infty}^{\infty} p(h'_{d,i,:}|\rho_{i,:}, x'_{d,i})p(\rho_{i,:})d\rho_{i,:}$ 
5     Update  $x_{d,i}^{(k)}$  and  $h_{d,i,:}^{(k)}$  to  $x_{d,i}^{(k+1)}$  and  $h_{d,i,:}^{(k+1)}$  with an acceptance
       probability  $\min \left\{ 1, \frac{p(\tilde{h}_{d,i,:}|h'_{d,i,:})}{p(\tilde{h}_{d,i,:}|h_{d,i,:})} \right\}$ 

```

---

density

$$\int_{-\infty}^{\infty} p(x_{d,i}|\lambda_i)p(\lambda_i)d\lambda_i,$$

since  $\lambda_i$  is distributed with some Gamma distribution and  $(x_{d,i}|\lambda_i)$  is distributed with Poisson distribution with parameters  $\lambda_i$ . Solving out the given integral in its general form for each  $x_{d,i}$  yields:

$$\begin{aligned}
\int_{-\infty}^{\infty} p(x_{d,i}|\lambda_i)p(\lambda_i)d\lambda_i &= \int_{-\infty}^{\infty} \prod_{i=1}^n \frac{\exp(-\lambda_i)\lambda_i^{x_{d,i}}}{x_{d,i}!} \frac{\alpha^\beta}{\Gamma(\beta)} \lambda_i^{\beta-1} \exp(-\alpha\lambda_i)d\lambda_i \\
&= \prod_{i=1}^n \int_{-\infty}^{\infty} \frac{\alpha^\beta}{x_{d,i}!\Gamma(\beta)} \lambda_i^{x_{d,i}+\beta-1} \exp(-(\alpha+1)\lambda_i)d\lambda_i \\
&= \prod_{i=1}^n \frac{\alpha^\beta}{x_{d,i}!\Gamma(\beta)} \frac{\Gamma(x_{d,i}+\beta)}{(\alpha+1)^{x_{d,i}+\beta}} \\
&= \prod_{i=1}^n \frac{\Gamma(x_{d,i}+\beta)}{\Gamma(x_{d,i}+1)\Gamma(\beta)} \left( \frac{\alpha}{\alpha+1} \right)^\beta \left( \frac{1}{\alpha+1} \right)^{x_{d,i}}.
\end{aligned}$$

This calculation shows that the marginal density of each  $x_{d,i}$  is a Negative Binomial distribution. Similarly if we propose each  $h_{d,i,:}$  from its marginal density

$$\int_{-\infty}^{\infty} p(h_{d,i,:}|\rho_{i,:}, x_{d,i})p(\rho_{i,:})d\rho_{i,:},$$

since  $\rho_{i,:}$  is distributed with a Dirichlet distribution and  $(h_{d,i,:}|\rho_{i,:}, x_{d,i})$  is distributed with Multinomial distribution, solving out the given integral for each  $i$ 'th row of  $h_d$  yields:

$$\begin{aligned} \int_{-\infty}^{\infty} p(h_{d,i,:}|\rho_{i,:}, x_{d,i}) p(\rho_{i,:}) d\rho_{i,:} &= \int_{-\infty}^{\infty} \frac{\Gamma(\sum_{j=1}^n \delta_{i,j})}{\prod_{j=1}^n \Gamma(\delta_{i,j})} \prod_{j=1}^n (\rho_{i,j})^{h_{d,i,j} + \delta_{i,j} - 1} d\rho_{i,j} \\ &= \frac{\Gamma(\sum_{j=1}^n \delta_{i,j})}{\prod_{j=1}^n \Gamma(\delta_{i,j})} \int_{-\infty}^{\infty} \prod_{j=1}^n (\rho_{i,j})^{h_{d,i,j} + \delta_{i,j} - 1} d\rho_{i,j} \\ &= \frac{\Gamma(\sum_{j=1}^n \delta_{i,j})}{\prod_{j=1}^n \Gamma(\delta_{i,j})} \frac{\prod_{j=1}^n \Gamma(\delta_{i,j} + h_{d,i,j})}{\Gamma(\sum_{j=1}^n \delta_{i,j} + h_{d,i,j})}. \end{aligned}$$

This density is known as the Multinomial-Dirichlet Compound distribution, hence the marginal density of  $h_{d,i,:}$  is a Multinomial-Dirichlet Compound distribution. It is then possible to propose both  $x_{d,i}$  and  $h_{d,i,:}$  from above mentioned marginal densities and accept them with the probability in relation 3.6. This modification allows us to omit the steps regarding updating  $\lambda_i$  and  $\rho_{i,:}$  since we are integrating over them.

### 3.3.4. Markov Chain Monte Carlo - Type 4

We have designed a one more type of proposal density utilizing the similar mentality behind the Type 3 proposal density. For this type of proposal density,  $x_{d,i}$  and  $h_{d,i,:}$  are proposed from proposal densities computed using their posteriors with Metropolis Hastings algorithm and updated according to the computed acceptance probability. Since we are directly proposing  $x_{d,i}$  and  $h_{d,i,:}$  from their posterior densities, it is not necessary to update and store  $\lambda_i$  and  $\rho_{i,:}$  values. Since these values are neither updated nor stored, the Gibbs step in which these parameters were updated is omitted and the overall steps of the Metropolis Hastings algorithm at iteration  $k$  is given by Algorithm 8.

The acceptance probability of this type of proposal density can be expressed as:

$$\min \left\{ 1, \frac{p(x'_{d,i}) p(h'_{d,i,:}|x'_{d,i}) q(x_{d,i}|x'_{d,i}) q(h_{d,i,:}|x_{d,i}, h'_{d,i,:}) p(\tilde{h}_{d,i,:}|h'_{d,i,:})}{p(x_{d,i}) p(h_{d,i,:}|x_{d,i}) q(x'_{d,i}|x_{d,i}) q(h'_{d,i,:}|x'_{d,i}, h_{d,i,:}) p(\tilde{h}_{d,i,:}|h_{d,i,:})} \right\}. \quad (3.14)$$



---

**Algorithm 8:** Metropolis Hastings Algorithm regarding Type 4 proposal density at iteration  $k$

---

```

1 for  $i = 1, \dots, n$  do
2   for  $d = 1, \dots, D$  do
3     Draw an  $x'_{d,i}$  from  $q(x'_{d,i}|x_{d,i}^{(k-1)}) = \int_{-\infty}^{\infty} p(\lambda_i|x_{d,i}^{(k-1)})p(x'_{d,i}|\lambda_i)d\lambda_i$ 
4     Draw an  $h'_{d,i,:}$  from
        $q(h'_{d,i,:}|x_{d,i}^{(k-1)}, h_{d,i,:}^{(k-1)}) = \int_{-\infty}^{\infty} p(h'_{d,i,:}|x'_{d,i}, \rho_{i,:})p(\rho_{i,:}|h_{d,i,:}^{(k-1)})d\rho_{i,:}$ 
5     Update  $x_{d,i}^{(k-1)}$  and  $h_{d,i,:}^{(k-1)}$  to  $x_{d,i}^{(k)}$  and  $h_{d,i,:}^{(k)}$  respectively with the
       computed acceptance probability

```

---

The above mentioned probability is easily computable since each of these densities correspond to computable distributions. For this type of proposal density, the  $\lambda_i$ 's and  $\rho_{i,:}$ 's are not stored and updated, instead the other densities are evaluated as an integration over of these parameters. Hence

$$p(x'_{d,i}) = \int_{-\infty}^{\infty} p(x'_{d,i}|\lambda_i)p(\lambda_i)d\lambda_i.$$

Since  $p(x'_{d,i}|\lambda_i)$  a Poisson density and  $p(\lambda_i)$  is a Gamma density, the integral is similarly to the Type 3 model solves out to be the probability mass function of a Negative Binomial density, furthermore this mechanic is called the Gamma-Poisson mixture. Similarly, since  $p(\lambda_i|x_{d,i})$  is a Gamma density and  $p(x'_{d,i}|\lambda_i)$  is a Poisson density,

$$q(x'_{d,i}|x_{d,i}) = \int_{-\infty}^{\infty} p(\lambda_i|x_{d,i})p(x'_{d,i}|\lambda_i)d\lambda_i$$

solves out to be the probability mass function of a Negative Binomial random variable as well. This computation can also be implemented for  $q(x_{d,i}|x'_{d,i})$ .

As for the conditional density of  $h_{d,i,:}$  given  $x_{d,i}$ , we have

$$p(h_{d,i,:}|x_{d,i}) = \int_{-\infty}^{\infty} p(\rho_{i,:})p(h_{d,i,:}|x_{d,i}, \rho_{i,:})d\rho_{i,:}$$

Since  $p(\rho_{i,:})$  is a Dirichlet distribution and  $p(h_{d,i,:}|x_{d,i}, \rho_{i,:})$  is a Multinomial distribution, this integral is a Multinomial-Dirichlet Compound distribution density. The calculation of the value of  $p(h'_{d,i,:}|x'_{d,i})$  then follows the same logic. Similarly, since  $p(h'_{d,i,:}|x'_{d,i}, \rho_{i,:})$  is a Multinomial density whereas  $p(\rho_{i,:}|h_{d,i,:})$  is a Dirichlet density

$$q(h'_{d,i,:}|x'_{d,i}, h_{d,i,:}) = \int_{-\infty}^{\infty} p(h'_{d,i,:}|x'_{d,i}, \rho_{i,:})p(\rho_{i,:}|h_{d,i,:})d\rho_{i,:}$$

solves out to be the density of a Multinomial-Dirichlet Compound distribution as mentioned in previously with Type 3 proposal density. This result can also be implemented on the calculation for  $q(h_{d,i,:}|x_{d,i}, \rho_{i,:})$ .

The calculation of  $p(\tilde{h}_{d,i,:}|h_{d,i,:})$  and  $p(\tilde{h}_{d,i,:}|h'_{d,i,:})$  terms in the acceptance probability can be computed same as the previous types of proposal densities. The aim of designing different proposal densities was to monitor the effect of the proposal density and to determine which type of proposal density is giving more realistic results. In Section 3.4, results regarding different types of proposal densities can be seen.

### 3.4. Discussion and Results

Since only two of the models generate samples of  $\rho$  and  $\lambda$ , calculation of the MSE values for each model is not possible, hence MSE is not a valid indicator of performance when comparing all models to each other. In order to decide which proposal density yields more effective samples, an analysis on the effective sample sizes of the samples created by each model was conducted. To conduct this analysis, calculation of the integrated auto correlation (IAC) times regarding samples collected from each model under different  $\epsilon$  values were made. The IAC time is calculated through the evaluation of the function:

$$f(\theta^{(k)}) = \sum_{d=1}^D \log p(X_d^{(k)}) + \log p(H_d^{(k)}|X_d^{(k)}) + \log p(\widetilde{H}_d|H_d^{(k)}) \quad (3.15)$$

where  $k$  represents the index of the sample collected from the  $k$ 'th iteration. Note that, since a burn-in period is introduced to the models, the calculation of the  $f$  values are conducted for the samples created after the burn-in period for a fair evaluation. As discussed in Section 3.3,  $p(X_d^{(k)})$  is a Negative Binomial density,  $p(H_d^{(k)}|X_d^{(k)})$  is a Dirichlet-Multinomial density, and  $p(\tilde{H}_d|H_d^{(k)})$  is Laplace density. The value of the function  $f$  is calculated for each sample created by the models and stored for IAC time calculation. We have investigated the models in two scenarios, where  $D = 1$  and  $D = 7$ . As  $D$  gets larger, the computation of the results get more computationally expensive. Therefore we have started our analysis by taking  $D = 1$  and calculated the IAC time values computed for each model under the  $\epsilon$  values chosen as:  $[0.1, 0.2, 0.5, 1, 2, 5, 10]$ . The values of IAC times calculated for this analysis can be seen in Table 3.2:

Table 3.2. Values of IAC times yielded by each model under different  $\epsilon$  values

$\epsilon$	Type 1	Type 2	Type 3	Type 4
0.01	1430.8	3411.3	4596.1	2998.7
0.02	2515.9	7212	7999	6641.3
0.05	7297.9	6703.1	2825	7304.7
1	4903.4	7169.1	7335.9	7516.5
2	6686.3	7380.9	5463.6	3734.9
5	7561	7576.7	4756.6	8147.9
10	6034.6	7366.5	4669.6	7529.4

The lower values of the IAC times are more desirable since a lower value of the IAC time means that the samples created by the given model are less correlated. Looking at the Table 3.2, the models that yielded the lowest and highest average IAC values are Type 1 and Type 2 respectively. However it is not possible to conclude that Type 1 model works more efficiently than the other types of models since the results correspond to a single run of MCMC and further analysis is required. In order to address that, we have decided to run models Type 1 and Type 2 for 15 times for  $D = 7$  under  $\epsilon = [0.1, 0.2, 0.5, 1]$ . The IAC values obtained for this scenario are given in Table 3.3.

Since both of these models generate direct samples of  $\rho$  and  $\lambda$ , the posterior mean

Table 3.3. Values of IAC times yielded by models in Scenario 2 under different  $\epsilon$ 

$\epsilon$	values	
	Type 1	Type 2
1	7714.9	7104.2
2	8000.7	7730.3
5	7980.4	7799.4
10	7780	7790.6

can be computed from these samples and can be compared to the true posterior mean of the  $\rho$  matrix and  $\lambda$  vector used to generate the noisy data. The calculation of the posterior mean of  $\rho$  for a given MCMC run is

$$\mu_{\rho_{i,j}} = \mathbb{E}[\rho_{i,j}|\tilde{H}] \approx \frac{1}{K - t_{\text{burn}}} \sum_{k=t_{\text{burn}}+1}^K \rho_{i,j}^{(k)} \quad (3.16)$$

and the calculation of the posterior mean of  $\lambda$  for a given run is:

$$\mu_{\lambda_i} = \mathbb{E}[\lambda_i|\tilde{H}] \approx \frac{1}{K - t_{\text{burn}}} \sum_{k=t_{\text{burn}}+1}^K \lambda_i^{(k)} \quad (3.17)$$

where  $K$  denotes the total number of samples collected and  $k$  denotes the index of the  $k$ 'th sample. The posterior mean generated by each MCMC run is then compared to the true posterior mean of the  $\rho$  matrix. The true posterior mean of  $\rho$ ,  $p(\rho|H)$ , can be calculated using the original values stored when the data was generated. As discussed in the Section 3.3, the posterior density  $p(\rho_{i,:}|h_{1,i,:}, \dots, h_{D,i,:})$  is a Dirichlet distribution with parameters  $(\delta_{i,:} + \sum_{d=1}^D h_{d,i,:})$ . Utilizing this information, the posterior density of a known Dirichlet distribution can be calculated. Similarly, the posterior density  $p(\lambda_i|x_{1,i}, \dots, x_{D,i})$  is a Gamma distribution with parameters  $(\alpha + \sum_{d=1}^D x_{d,i})$  and  $(\beta + D)$ . The norm of the difference vector of  $\rho$  and  $\lambda$ , defined as the difference between the true posterior mean and the posterior mean generated by the MCMC samples of  $\rho$  and  $\gamma$  respectively, were calculated for each MCMC run of models Type 1 and Type 2. The calculated mean value of norm of the difference of  $\rho$  and  $\lambda$  samples generated by the MCMC runs after the burn in period with respect to corresponding  $\epsilon$

values are given in Table 3.4 and Table 3.5.

Table 3.4. Mean Value of Norm of the Difference Matrix of  $\rho$  for Type 1 and Type 2

$\epsilon$	Mean Value of Norm for Type 1	Mean Value of Norm for Type 2
1	0.0041	0.0040
2	0.0034	0.0032
5	0.0029	0.0028
10	0.0028	0.0027

Table 3.5. Mean Value of Norm of the Difference Matrix of  $\lambda$  for Type 1 and Type 2

$\epsilon$	Mean Value of Norm for Type 1	Mean Value of Norm for Type 2
1	7038	7033
2	7040	7039
5	7032	7038
10	7035	7038

It is observed that both of the mean of the norm of the difference vector calculated for  $\rho$  decreases as the  $\epsilon$  value increases, therefore it can be concluded that the models generate samples with smaller error as the noise decreases. The mean value of the norm of the difference vector calculated for  $\lambda$ 's for both models do not change according to the value of  $\epsilon$ . These results are consistent with each other since the values calculated are approximately equal. The trend of the decrease in the mean value of the norm of the difference for  $\rho$  can be explained with the generated data being more noisy as the  $\epsilon$  value increases and hence the accuracy of the algorithm slightly decreases. We would also expect this result since the data becomes more random as the noise increases, hence the algorithm yields a higher error. In Figure 3.1, Figure 3.3, Figure 3.5, and Figure 3.7, samples of  $\lambda$ 's and in Figure 3.2, Figure 3.4, Figure 3.6, and Figure 3.8 samples of  $\rho$  generated by model Type 1 during a randomly selected MCMC run under different values of  $\epsilon$  can be seen. Moreover, In Figure 3.9, Figure 3.11, Figure 3.13, and Figure 3.15, samples of  $\lambda$ 's and in Figure 3.10, Figure 3.12, Figure 3.14, and Figure 3.16 samples of  $\rho$  generated by model Type 2 during a randomly selected MCMC run under different values of  $\epsilon$  can be seen. In addition to the results we obtained from the norm analysis, these figures further prove that both of the models perform efficiently and yield realistic results when estimating the parameters  $\lambda$  and  $\rho$ .

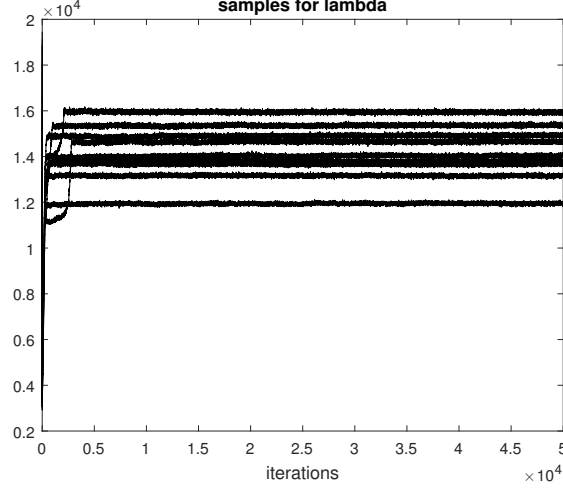


Figure 3.1. Comparison of the  $\lambda$  samples generated by model Type 1 with their true value through iterations when  $\epsilon = 1$

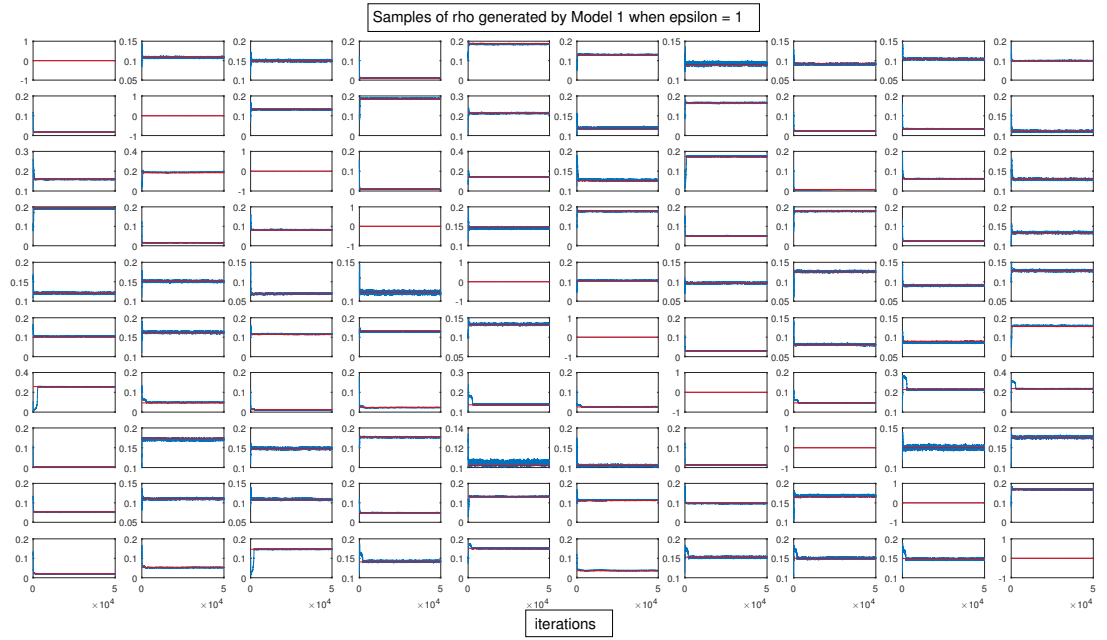


Figure 3.2. Comparison of the  $\rho$  samples generated by model Type 1 with their true values for  $\epsilon = 1$

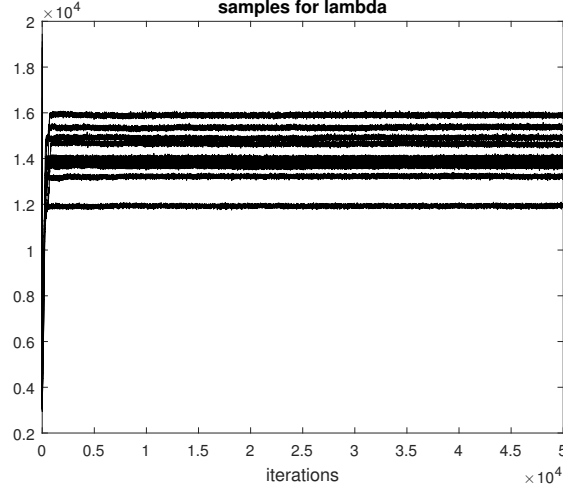


Figure 3.3. Comparison of the  $\lambda$  samples generated by model Type 1 with their true value through iterations when  $\epsilon = 2$

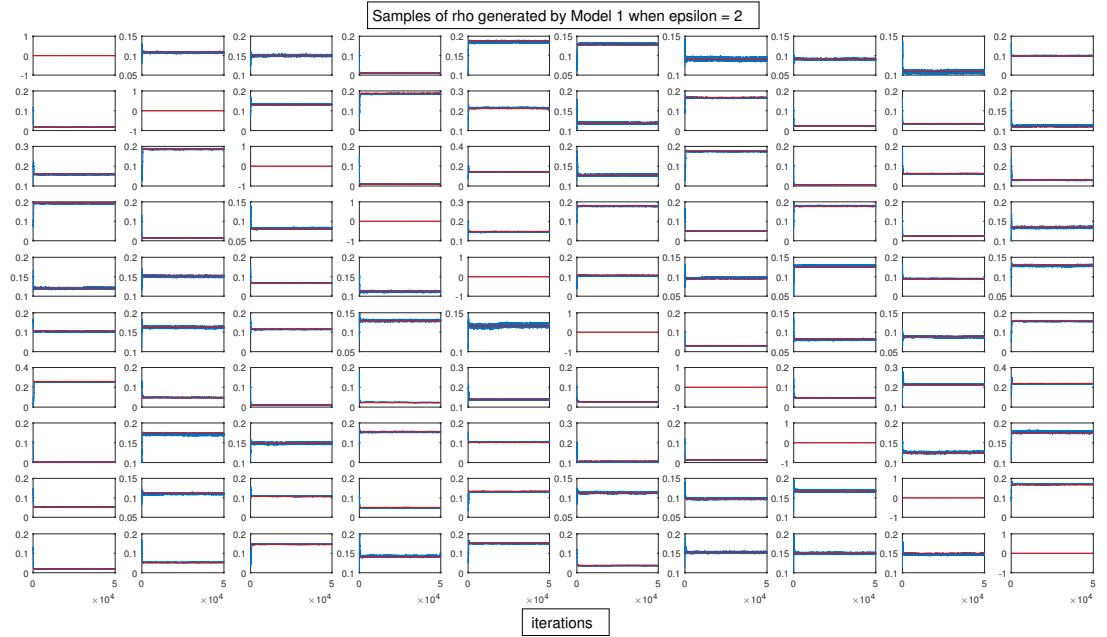


Figure 3.4. Comparison of the  $\rho$  samples generated by model Type 1 with their true values for  $\epsilon = 2$

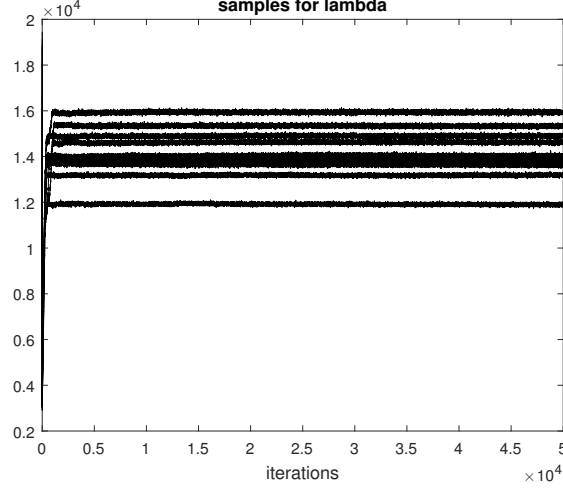


Figure 3.5. Comparison of the  $\lambda$  samples generated by model Type 1 with their true value through iterations when  $\epsilon = 5$

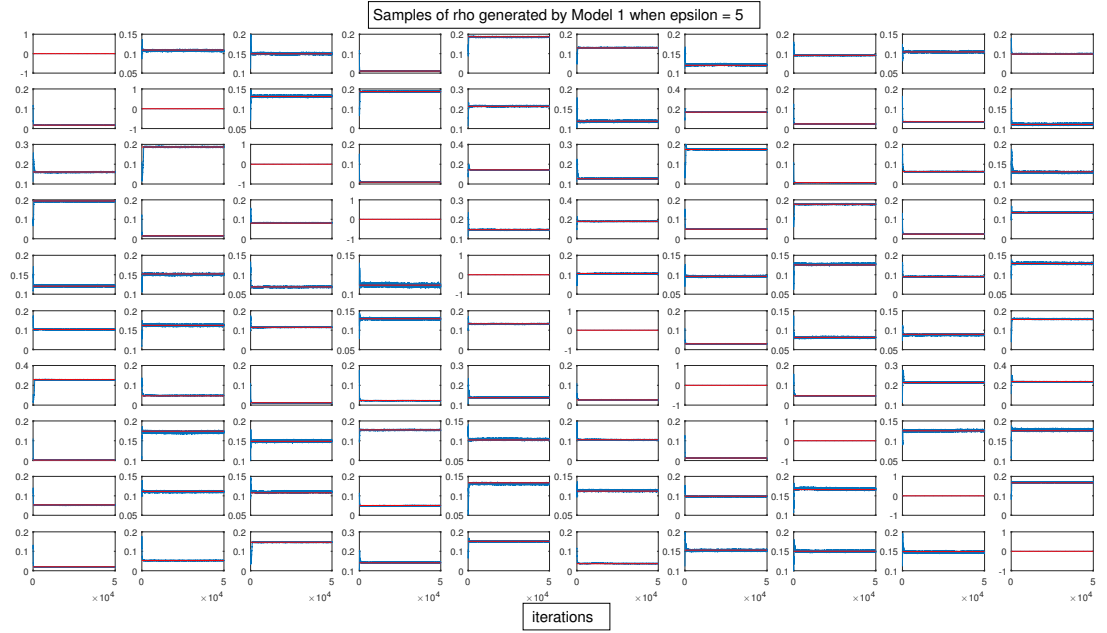


Figure 3.6. Comparison of the  $\rho$  samples generated by model Type 1 with their true values for  $\epsilon = 5$



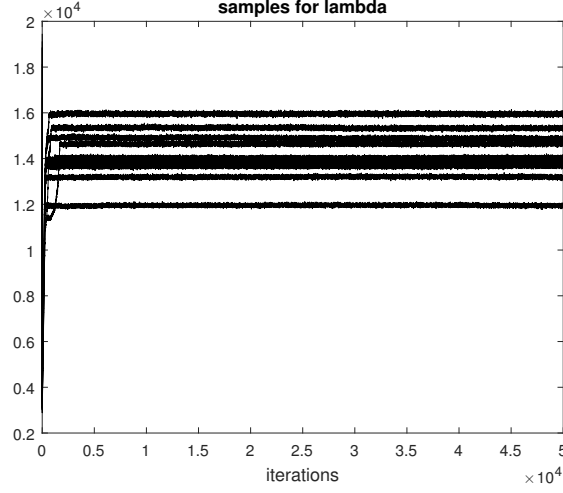


Figure 3.7. Comparison of the  $\lambda$  samples generated by model Type 1 with their true value through iterations when  $\epsilon = 10$

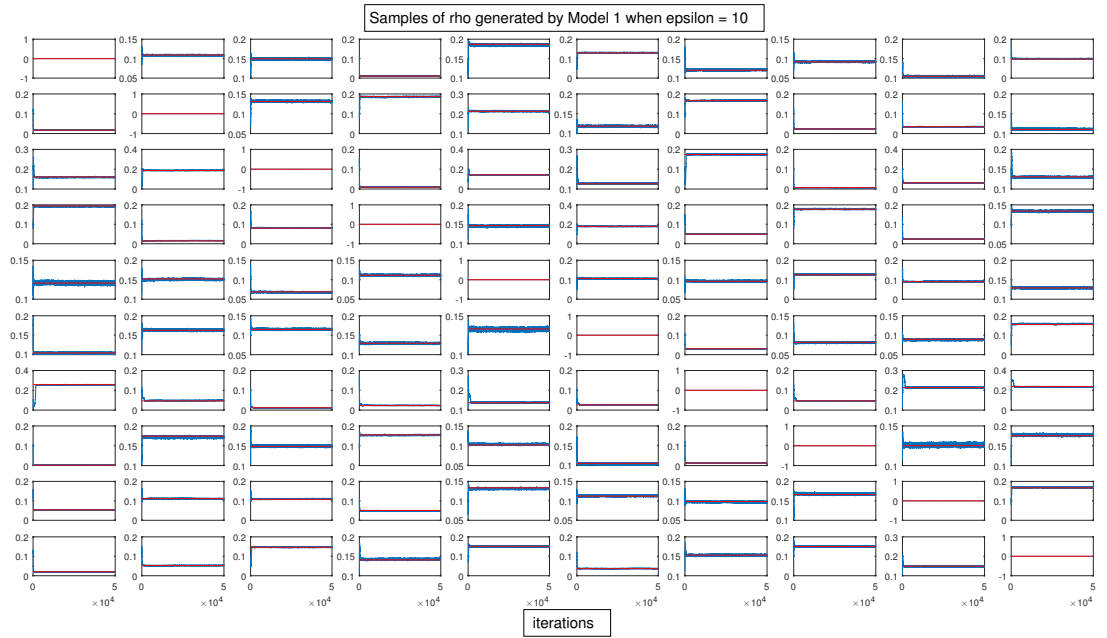


Figure 3.8. Comparison of the  $\rho$  samples generated by model Type 1 with their true values for  $\epsilon = 10$

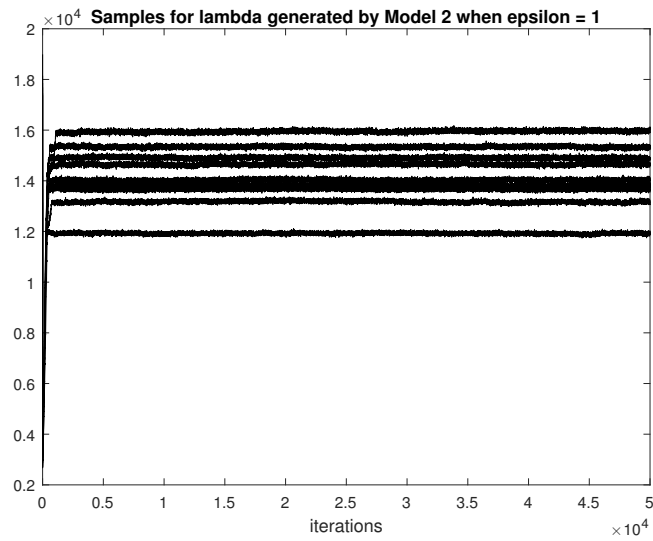


Figure 3.9. Comparison of the  $\lambda$  samples generated by model Type 2 with their true value through iterations when  $\epsilon = 1$

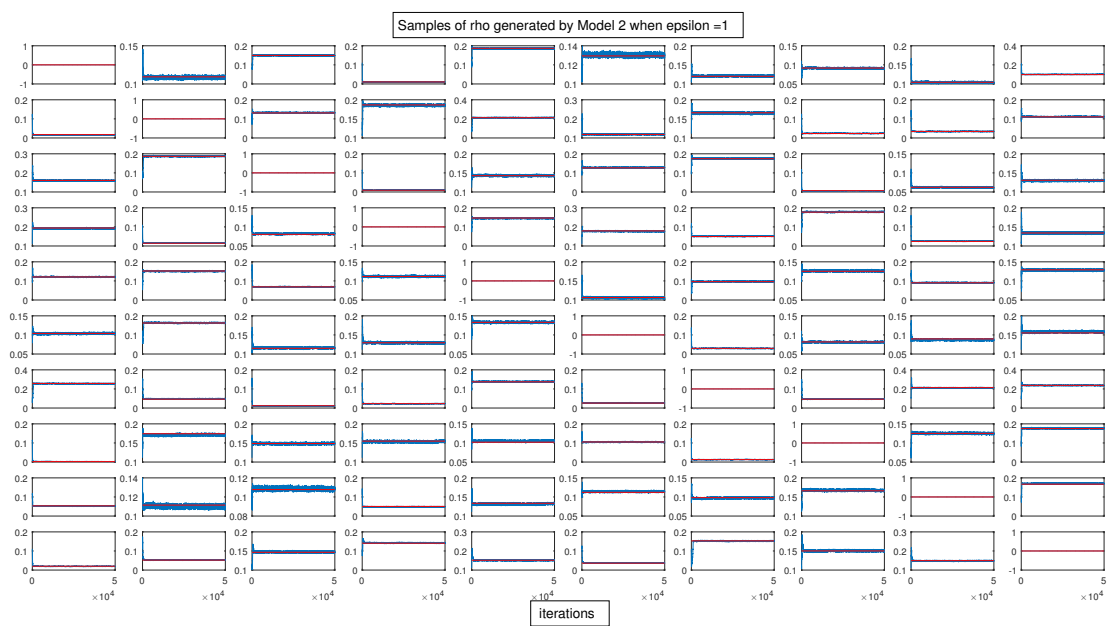


Figure 3.10. Comparison of the  $\rho$  samples generated by model Type 2 with their true values for  $\epsilon = 1$

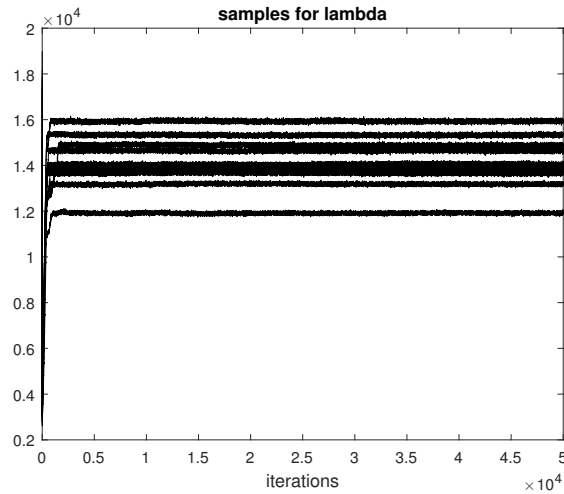


Figure 3.11. Comparison of the  $\lambda$  samples generated by model Type 2 with their true value through iterations when  $\epsilon = 2$

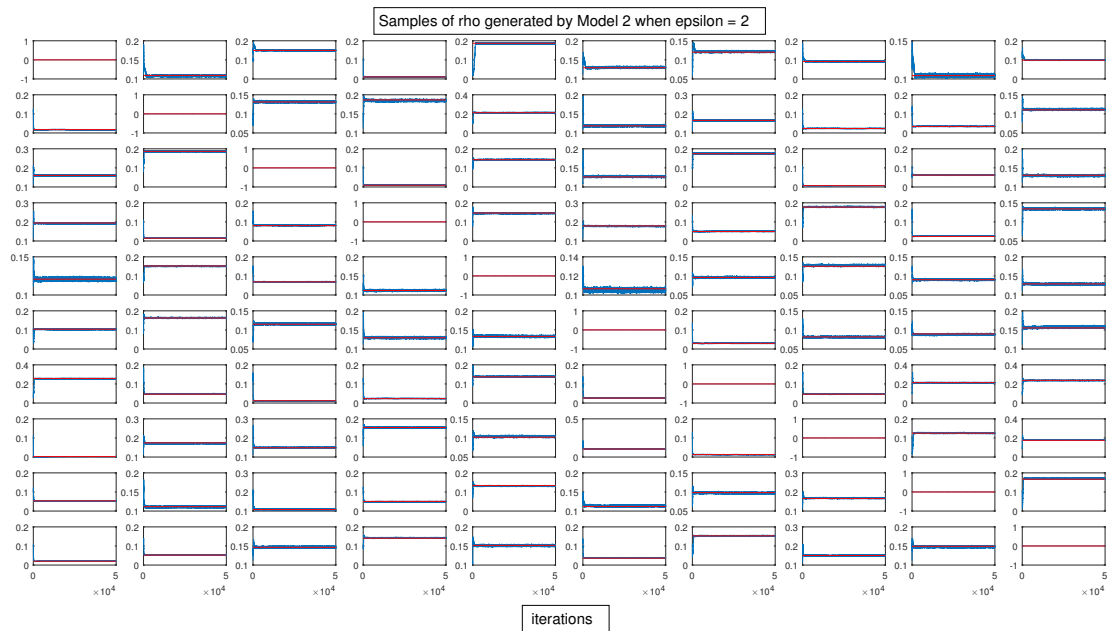


Figure 3.12. Comparison of the  $\rho$  samples generated by model Type 2 with their true values for  $\epsilon = 2$

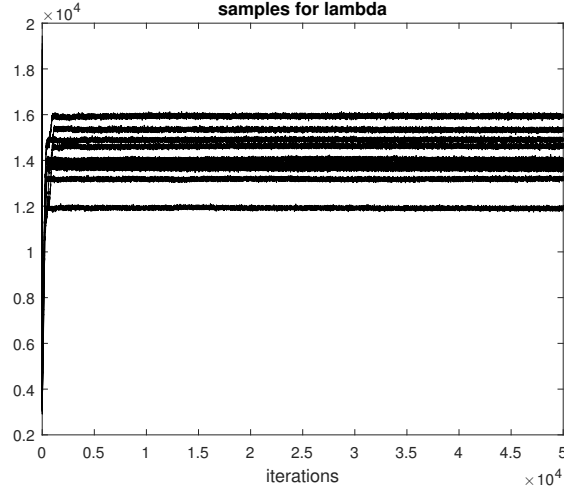


Figure 3.13. Comparison of the  $\lambda$  samples generated by model Type 2 with their true value through iterations when  $\epsilon = 5$

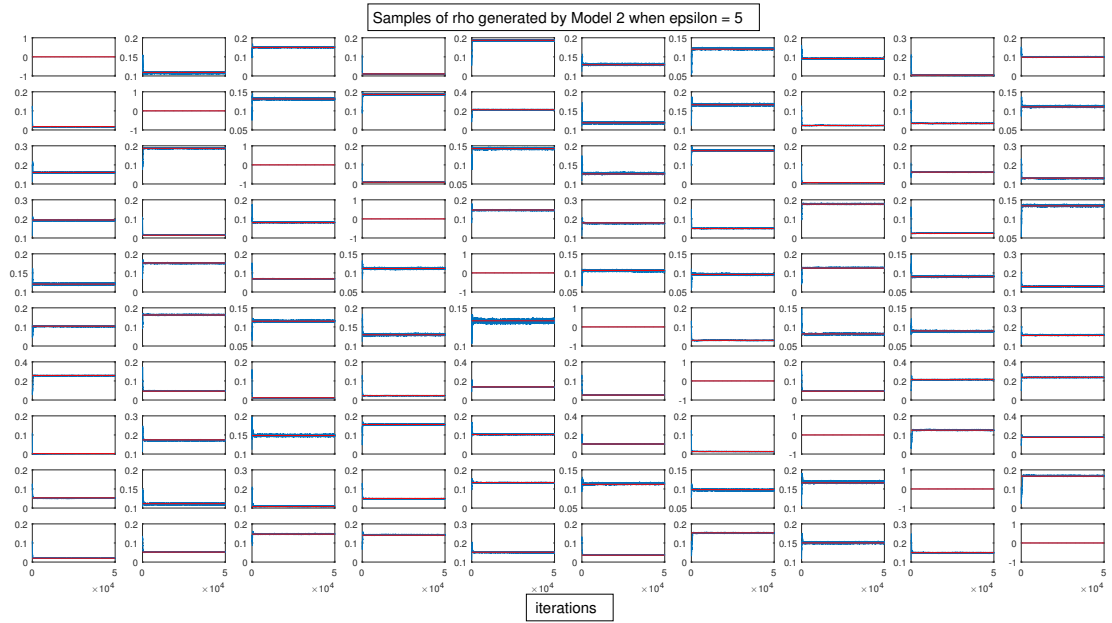


Figure 3.14. Comparison of the  $\rho$  samples generated by model Type 2 with their true values for  $\epsilon = 5$

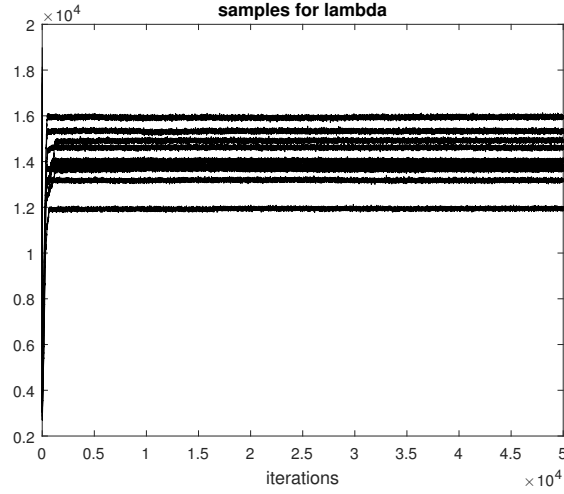


Figure 3.15. Comparison of the  $\lambda$  samples generated by model Type 2 with their true value through iterations when  $\epsilon = 10$

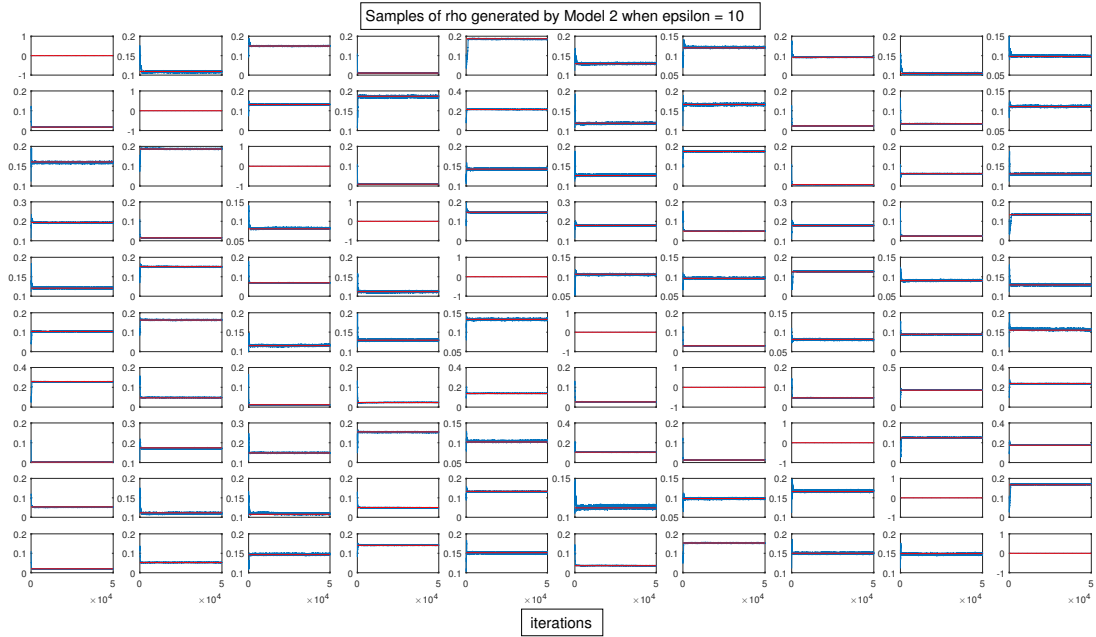


Figure 3.16. Comparison of the  $\rho$  samples generated by model Type 2 with their true values for  $\epsilon = 10$

## 4. CONCLUSION

The main aim of this research was to propose a novel method of estimation for parameters of the Origin-Destination problem using the of Markov chain Monte Carlo methods, Gibbs Algorithm, Metropolis-Hastings algorithm, and Metropolis-Hastings within Gibbs algorithm.

In Chapter 2, a probability model for the Missing Data scenario was formulated and the Metropolis-Hastings within Gibbs algorithm was utilized. For this scenario, it was assumed, that the data obtained from the data holders were the original data, but with a missing column of departure information. The algorithm used to estimate the parameters  $\alpha$  and the  $\rho$  matrix of the origin-destination matrix was proven to be a viable methodology since it can estimate and generate samples which are very close to the true values with small variation as discussed in Section 2.5. This conclusion means that, if the original data follows the assumptions of our model and the synthetically generated data, then we would also be able to estimate these values for the original data. Improvements which can be conducted in the future include the computation of different variations of the probability model. For example, if we have a reason to believe that stations closer to the arrival station are more likely to be the departure station, we can update our probability model as:

$$g_{\alpha}(b|d, \tau) = \frac{\exp \left\{ -\alpha \frac{|b-d|}{\tau} \right\}}{\sum_{k=1}^n \exp \left\{ -\alpha \frac{|k-d|}{\tau} \right\}}. \quad (4.1)$$

In our future work, we will also explore different models, which might more realistically represent the original data. Alterations such as the one mentioned in 4.1 might be taken into consideration. Another improvement could be selection of different proposal densities and comparing and contrasting the performance of the algorithm when different proposal densities are chosen. As it was demonstrated in Chapter 2, the selection of the proposal density has great effects on the effectiveness of the MCMC algorithm.

In Chapter 3, 4 types of MCMC algorithms differing in proposal densities were formulated for the Noisy Data scenario and their performance regarding the related measures were compared. In contrast with the scenario discussed in Chapter 2, for this scenario it was assumed, that the whole data was collected, but the data holder did not release the original data, instead provided a version that was protected with Laplace mechanism. These algorithms proved utilization of Markov chain Monte Carlo methods to be a viable method to estimate the parameters the  $\lambda$  vector and the  $\rho$  matrix of an origin-destination problem when the obtained data is protected with Laplace mechanism due to data privacy issues. However the choice of proposal density plays the most important role in the performance of the model as the results indicated great variability in the performance as the proposal density varies. It is also important to note that, it was found that the performance of the MCMC algorithm decreased slightly as the data became more noisy. The improvements which can further enhance the performance of the algorithms would be to select a proposed density for the Metropolis Hastings move which demonstrates higher performance on the mentioned criteria, therefore a more realistic approach. We were not able to obtain the data from the related government office, hence we have worked with the synthetically generated data. Another improvement for our future work would be obtaining the original data and measuring the performance of the algorithms presented in both Chapter 2 and Chapter 3 when the original data is used. Replicating these results with the original data would further validate the results we obtained using the synthetic data.

Our work contributes to the origin destination matrix estimation studies in a way that was not approached before. The previous work in the literature related to the Missing Data scenario provides estimations without a Markovian approach, either through passenger, surveys which are costly and dependent on passenger honesty for their validity, or through counting approaches without statistical flavour, or through usage of other means of data such as mobile phone location data (Calabrese et al., 2011). Ni and Leonard II (2005) proposed that Markov chain Monte Carlo methods can be used to impute values in a missing data environment so that they have used MCMC algorithms to propose values for the missing data points, and we have improved this approach by further using an MCMC algorithm that also updates the parameters in the

probability model in order to estimate the  $\rho$  matrix of the origin-destination problem. As it was mentioned in Chapter 1, there was not yet a published study conducted at the time of this thesis was written on the performance of the Markov chain Monte Carlo methods for parameter estimation in a differentially private data environment for the Origin-Destination problem however MCMC algorithms were found to be a possible viable option for parameter estimation for differentially private data (Lu and Miklau, 2014). To the best of our knowledge, the approach discussed in this thesis is novel.



## REFERENCES

- Calabrese, F., Di Lorenzo, G., Liu, L., and Ratti, C. (2011). Estimating origin-destination flows using opportunistically collected mobile phone location data from one million users in boston metropolitan area. *IEEE Pervasive Computing*, 10(4):36–44.
- Cascetta, E., Inaudi, D., and Marquis, G. (1993). Dynamic estimators of origin-destination matrices using traffic counts. *Transportation Science*, 27(4):363–373.
- Cascetta, E. and Nguyen, S. (1988). A unified framework for estimating or updating origin/destination matrices from traffic counts. *Transportation Research Part B Methodological*, 22B(6):437–455.
- Charest, A.-S. (2010). How can we analyze differentially-private synthetic datasets? *Journal of Privacy and Confidentiality*, 2(2):21–33.
- Chen, R., Desai, C. B., Fung, C. B., and Sossou, M. N. (2014). Differentially private transit data publication: A case study on the montreal transportation system. *Foundations and Trends in Theoretical Computer Science*, 9:211–407.
- Dwork, C. and Roth, A. (2014). The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9:211–407.
- Gelfand, A. and Smith, A. (1990). Sampling-based approaches to calculating marginal densities. *Journal of the American Statistical Association*, 85(410):398–409.
- Geman, S. and Geman, D. (1984). Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 6(6):721–741.
- Hastings, W. K. (1970). Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 52(1):97–109.
- Hazelton, Martin, L. (2001). Inference for origin-destination matrices: Estimation, prediction, and reconstruction. *Transportation Research Part B Methodological*, 35:667–676.

- Li, B. (2005). Bayesian inference for origin-destination matrices of transport networks using the em algorithm. *Technometrics*, 47(4):399–408.
- Lu, W. and Miklau, G. (2014). Exponential random graph estimation under differential privacy. Technical report, University of Massachusetts Amherst.
- Maher, M. J. (1983). Inferences on trip matrices from observations on link volumes: A bayesian statistics approach. *Transportation Research Part B Methodological*, 17B(6):435–447.
- Metropolis, N. and Ulam, S. (1949). The monte carlo method. *Journal of the American Statistical Association*, 44(247).
- Munizaga, A. M., Devillaine, F., Navarrete, C., and Silva, D. (2014). Validating travel behavior estimated from smartcard data. *Transportation Research Part C*, 44:70–79.
- Munizaga, A. M. and Palma, C. (2012). Estimation of a disaggregate multimodal public transport origin–destination matrix from passive smartcard data from santiago, chile. *Transportation Research Part C*, 24:9–18.
- Ni, D. and Leonard II, D. J. (2005). Markov chain monte carlo multiple imputation using bayesian networks for incomplete intelligent transportation systems data. *Transportation Research Record: Journal of the Transportation Research Board*, (1935):57–67.
- Tebaldi, C. and West, M. (1998). Bayesian inference on network traffic using link count data. *Journal of the American Statistical Association*, 93:557–573.
- Watling, P. D. (1994). Maximum likelihood estimation of an origin-destination matrix from a partial registration plate survey. *Transportation Research Part B Methodological*, (4):289–314.
- Yıldırım, S. (2016). Simulation methods for statistical inference. Sabanci University Class Notes.
- Zhao, J. and Rahbee, A. (2007). Estimating a rail passenger trip origin-destination matrix using automatic data collection systems. *Computer-Aided Civil and Infrastructure Engineering*, (22):376–387.